COST ESTIMATION TOOL FOR SOFTWARE DEVELOPMENT PROJECTS

A CASE STUDY OF UNICAL COMPUTER CENTRE (SOFTWARE DEVELOPMENT UNIT)

BY

OMAGU DAVID ABENNE

PGD/MCS/2007/1216

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGER STATE.

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF POST GRADUATE DIPLOMA IN COMPUTER SCIENCE.

APRIL, 2009.

DECLARATION

This project is an original work carried out by me under the supervision of **Dr. L. N Ezeako** and to the best of my knowledge, no part thereof has been submitted elsewhere for the award of any degree but works of others had been acknowledged and referenced.

OMAGU DAVID ABENNE

PGD/MCS/2007/1216

CERTIFICATION

This is to certify that this project work titled; **"Cost Estimation Tool for Software Development Projects"** A Case Study of Unical Computer Centre (Software Development Unit), was carried out by **OMAGU DAVID ABENNE**, Reg. **No.PGD/MCS/2007/1216** of the Mathematics/Computer Science Department, Federal University of Technology, Minna, and satisfies the requirement for the award of Postgraduate Diploma in Computer Science of the Federal University of Technology ,Minna.

______________________	07-05-2009
Dr. L.N. Ezeako	**Date**
(Supervisor)	
______________________	______________
Dr. N. I. Akinwande	**Date**
(Head of Department)	
______________________	______________
External Examiner	**Date**

DEDICATION

This project is dedicated to the ALMIGHTY GOD whose grace saw me through most of the challenges I encountered in the course of study, my elder brother, **Dr. Omagu Donald Omagu** and my elder Sister, **Mrs. Angela Agbo Otogo**

ACKNOWLEDGEMENT

I wish to express my profound gratitude to my project Supervisor, **Dr. L. N. Ezeako** for his patience, understanding, guidance and fatherly advice and also my academic adviser (our cordinator) **Mal A Ndanusa** for his advice, assistance and for ensuring a successful completion of the programme.

I also wish to express my gratitude to my parent's **Mr. and Mrs. John O. Omagu**, my sponsors, **Dr Omagu Donald Omagu, Mrs. Angela Agbo Otogo** and Lt Col Godwin Otogo for their encouragement and financial assistance.

I extend my heart felt gratitude to **Professor M. S Abolarin, Mr. Tunji Wahab, Mr. Falaye Adeyinka**, for the assistance and inspiration I received from them. I also acknowledge my lecturers Mal **Jiya, Mal Yakubu Yisa, Mr Onotu Idris, Mal Nafiu, Mal. A Usman, Mal saliu, Mr Gbolahan**, and a host of others.

I equally acknowledge my colleagues, **Sunny, Makun, Mrs Felix, Mr Fidelis, Sesan, Patrick, Yemi , Victor, Ruth, Lara and Niyi.**

To you all I say a **"BIG THANK YOU"**

ABSTRACT

One of the greatest challenges for software project leader is to successfully deliver on all aspects of a project both according to the client's specifications and within the allotted budget. More software projects have failed due to lack of proper estimation of cost at the outset of the project than any other reason. Our estimating techniques are generally poor and given low priority. When the software estimation process is performed accurately, the benefits realized far outweigh the cost of doing the estimation. Some of the major benefits include lowering the cost of doing business, increasing the probability of winning new contracts, increasing and broadening the skill – level of key staff members. As a result of this, there is need to focus some effort on improving the situation by developing a reliable and efficient tool to estimate the cost of developing software. Thus, the aim of this project is to provide a software tool to accurately give the estimated cost of developing a software project.

TABLE OF CONTENTS

CHAPTER FIVE

SUMMARY, RECOMMENDATIONS AND CONCLUSION

LIST OF TABLES

LIST OF FIGURES

In many cases estimates are made using past experiences as the only guide.

If a new project is quite similar in size and function to a past project, it is likely that the new project will require approximately the same amount of effort, time and cost as the last work.

But what if the project breaks new grounds? Then past experience alone may not be enough.

Is it possible to apply mathematical and scientific principles to software estimation, so that development schedules, cost ,productivity and quality might be objectively ascertained or estimated rather than being a matter of opinion? The answer to this question will be unraveled as we progress in this research work.

Hence, it is for this purpose that this research work is embarked upon to develop a tool for estimating cost for software development projects.

1.1 INTRODUCTION

Estimating is the process of forecasting a future result in terms of cost, based upon information available at the time. Industries have limited resources of personnel, time and budget and proper estimating will allow the leaders of the organization to properly allocate these limited resources to achieve the highest benefits. Hence, estimating is an invaluable tool for anticipating project uncertainties and managing these limited resources. Over the years many have attempted to determine apriori what the cost of developing a specific application will be. Why has it been so important? Not only is the budget on the line, but many times a manager's job or reputation as well. When we can identify our cost and schedule requirements with relative precision, it reduces the risk of running out of time, resources, and budget during a project .Also, when the software estimation process is performed correctly, the benefits realized far outweigh the cost of doing the estimation. Some of the major benefits include lowering the cost of doing business, increasing the probability of winning new contracts, increasing and broadening the skill –level of key staff members, acquiring a deeper knowledge of the proposed project prior to beginning the software development.

However, most software industry doesn't estimate projects well and does not use estimates appropriately .We suffers far more than we should as a result and we need to focus some effort on improving the situation.

Under –estimating a project leads to understaffing it (resulting in staff burnout), under-scoping the quality assurance Effort (running the risk of low quality deliverables) and setting too short a schedule (resulting in loss of credibility as deadlines are missed).For those who figure on avoiding this situation by generally padding the estimate, over estimating a project can be just about as bad for the organization. If you give a project more resources than it really needs without sufficient scope controls it will use them. The project is then likely to cost more than the it should (a negative impact on the bottom line), take longer to deliver than necessary (resulting in lost opportunities) and delay the use of your resources on next project.

1.2 Objectives of the Study

Since the origin of software development, the industry had been threatened by poor performance as a result of late delivery date, poor reliability, over budget and inability to estimate project size correctly. Even today, few people expect software to be delivered on time and within budget. It is for this reason that this project is embarked upon to research on ways of improving the industry, especially how much it will take to develop a new software project.

1.3 Scope of the study

This project scope is limited to the development of a software package that will estimate the cost of developing a new software project using the COCOMO(Constructive Cost Model)principle and SLIM Model .I intend to develop the package using an object oriented programming language such as visual Basic 6.0

1.4 Limitation of study

There are several limiting factors to this research work and some of them are as follows:

(i) Due to time constraints and dearth of materials the research was limited to just an aspect as discussed .

(ii) The other limitation in this work is lack of sufficient finance to speed up the research process.

(iii) As a result of the scarcity of software industry satisfactory data were not collected.

(iv) The secrecy with which the software house handles data imposed difficulty in data collection.

1.5 Definition of Terms

CMM: The capability maturity model (CMM) is a process maturity framework developed by the software Engineering Institute (SEI) to help organization improve their software processes.

KSLOC: One thousand lines of code and is a measurement of program size.

SLOC: Is the source lines of code (or lines of source code) and is a measurement of program size.

PERSON_MONTHS: A person –months (PM) is the amount of time one person spends working on the software project for one month.

COCOMO: The Constructive Cost Model (COCOMO) is the most widely used software estimation model in the world .The COCOMO model predicts the effort and duration of a project based on inputs relating to the size of the resulting systems and a number of "cost drives" that affect productivity .It is a model that allows one to estimate the cost, effort and schedule when planning a new software development activity.

PRODUCTIVITY: Productivity is the inverse of the labour rate or rather, the number of size units that can be put through an activity with a given effort.

Green (1991) asserted that there is still tremendous pressure for low **"Estimates"** during the bidding process by firms to do what it takes to secure new projects, in spite of the long and significant history of software projects that are either unable to deliver the expected benefits or simply cancelled. Reasons include lateness, exceeded budgets, unacceptable low quality or some combination of these problems.

Continuing with the history of cost estimation, in 1983, Dr Howard Rubin's ESTIMACS model reached the commercial market. ESTIMACS was a derivative of some of IBM's internal estimating methodology for information systems project and supported an early form of function point metrics prior to IBM's major revision of function points in 1984, which is the basis of today's standard function point. In 1985 the author's SPQR/20(for Software productivity, quality, and reliability) estimation tool reached the commercial market. The SPQR/20 was the first commercial software estimator built explicitly to support function metrics. It was also the first software estimation tool to include full sizing logic, and the first to include "backfiring" or direct conversion from lines of code (LOC) metrics to function point metrics and vice versa. The tool was also the first to integrate software cost estimation, software risk analysis and software quality estimation in a single tool. In the last decade until this year 2009 researches in Software Cost Estimation become numerou side to side with the rapid exponential improvement in software and information technology industry, mainly the researches related to handle the new phenomena in software engineering such as object oriented environment, real time systems etc via new sizing techniques, some new measures to measure the new existed items and improving and calibrating the previous models and techniques to be applicable to the new environment.

2.2 History of Unical Computer Centre

Unical (University of Calabar) Computer Centre was established and commissioned by Professor A B Fafunwa ,Pro- Chancellor and Chairman of Council, the University of Calabar on 29th of September 1988. The director appointed then was Mr. O .U Uchendu. The sole work of the computer centre the was to prepare/compute salary and account posting for bursary department using a SUR machine .Also, another function of the centre was to create computer awareness. The machine used then was Amstrad machine.

In 1991, Professor Lipcsey took over the leadership which brought changes such as the running of courses and a business centre.

In 2000, Mr. Ushie Godwin Ikwun took over the leadership with improvements in the services rendered including software development and data analysis.

2.3 Trends in the Centre

Since the inception of the centre till date, there has been a tremendous improvement and consolidation in the services rendered. The following are the services rendered:

(i) Training of students in applications such as Electronic Office Administration, Data processing, Programming, Desktop publishing, E-mail/ Internet operation and Browsing and Basic hardware maintenance.
(ii) Software development
(iii) Research data analysis in computerization including networking.
(iv) E-mail services
(v) Desktop publishing
(vi) Word processing
(vii) Computer Maintenance and repairs
(viii) Book – Publishing
(ix) Student's project and many more.

CHAPTER THREE

SYSTEM DESIGN METHODOLOGY

3.0 Introduction

Research methodology is a set of orderly logical procedures followed in carrying out a research work. The aim of this chapter is to analyze the old system and to develop a new system to overcome the problems encountered in the previous system thereby improving the services of your clients. It also involves the transformation of the needs uncovered in the analyses of the old system into possible ways of meeting them.

Software development is the process of changing the client's need into a product that satisfies those needs .The product is a set of interrelated objects or entities that are seen as a whole and designed to achieve a purpose.

3.1 Requirements

These are classified into User's requirements and the system requirements.

3.1.1 User's Requirement

The User's requirements include:

The system should be able to compute the size of the project.
The system should be capable of computing estimated effort and schedule.
The system should be capable of computing the cost of the project.
The system should be capable of computing the time and effort needed to carry out feasibility study and architectural design.
The system should be able to compute the size of the user's manual.

Expert Judgment

In this method, several expert on the proposed software development technique and the application domain are consulted .They each estimate the cost of the project .These estimate is compared and discussed and the estimation process iterates until an agreed estimate is reached.

Parkinson's Law

Parkinson's law states that work expands to fill the time available. The cost is determined by available resources rather than by objective assessment.

Pricing to Win

In this method, the software cost is estimated to be whatever the customer has available to spend on the project. The estimated effort depends on the customer's budget and not on the software functionality.

Estimation by Analogy

This technique is applicable when other projects in the same application domain have been completed. The cost of a new project is estimated by analogy with these completed projects.

3.3.2 Software Cost Estimation Procedures

The four basic steps in software estimation are;

Estimate the size of the development product. This generally ends up in either lines of code (LOC) or Function Points(FP)
Estimate the staff effort in person-months or person –hours
Estimate the schedule in calendar months.
Estimate the project cost in dollar(or local currency)

(1) Estimating Size.

An accurate estimate of the size of the software to be built is the first step to an effective estimate. Your source(s) of information regarding the scope of the project should wherever possible start with formal descriptions of the requirements. For example, a customer's requirement specification, a system specification, a software requirement s specification.

Two main ways of estimating product size are:

By analogy
By counting product features and using an algorithmic approach such as function points to convert the count into an estimate of size.

(2) Estimating Effort

Once you have an estimate of the size of your product, you can derive the effort estimate. This conversion from software size to total project effort can only be done if you have a defined software development lifecycle and development process that you follow to specify, design, develop and test the software. There are two main ways to derive effort from size:

(i)By analogy

(ii) By algorithmic approach

(3) Estimating Schedule

The third step in estimating a software development project is to determine the project schedule from the effort. This generally involves estimating the number of people who will work on the project, what they will work on, when they will start working on the project and when they will finish. Once you have this information, you need to lay it out into a calendar schedule. Schedule can be derived from effort by (i)Analogy

(ii) Algorithmic approach

(4) Estimating Cost

Once schedule have been estimated, it is easy to estimate cost either by analogy or algorithmic approach.

3.4 System Design

System design involves the transformation of the requirement disclosed in the analysis into possible ways of satisfying them. After we conclude what we are going to develop from the requirement specification phase, we then decide on how the development will take place in the design stage. System design involves two levels of operation:

Logical design and Physical design

3.4.1 Computer System Features

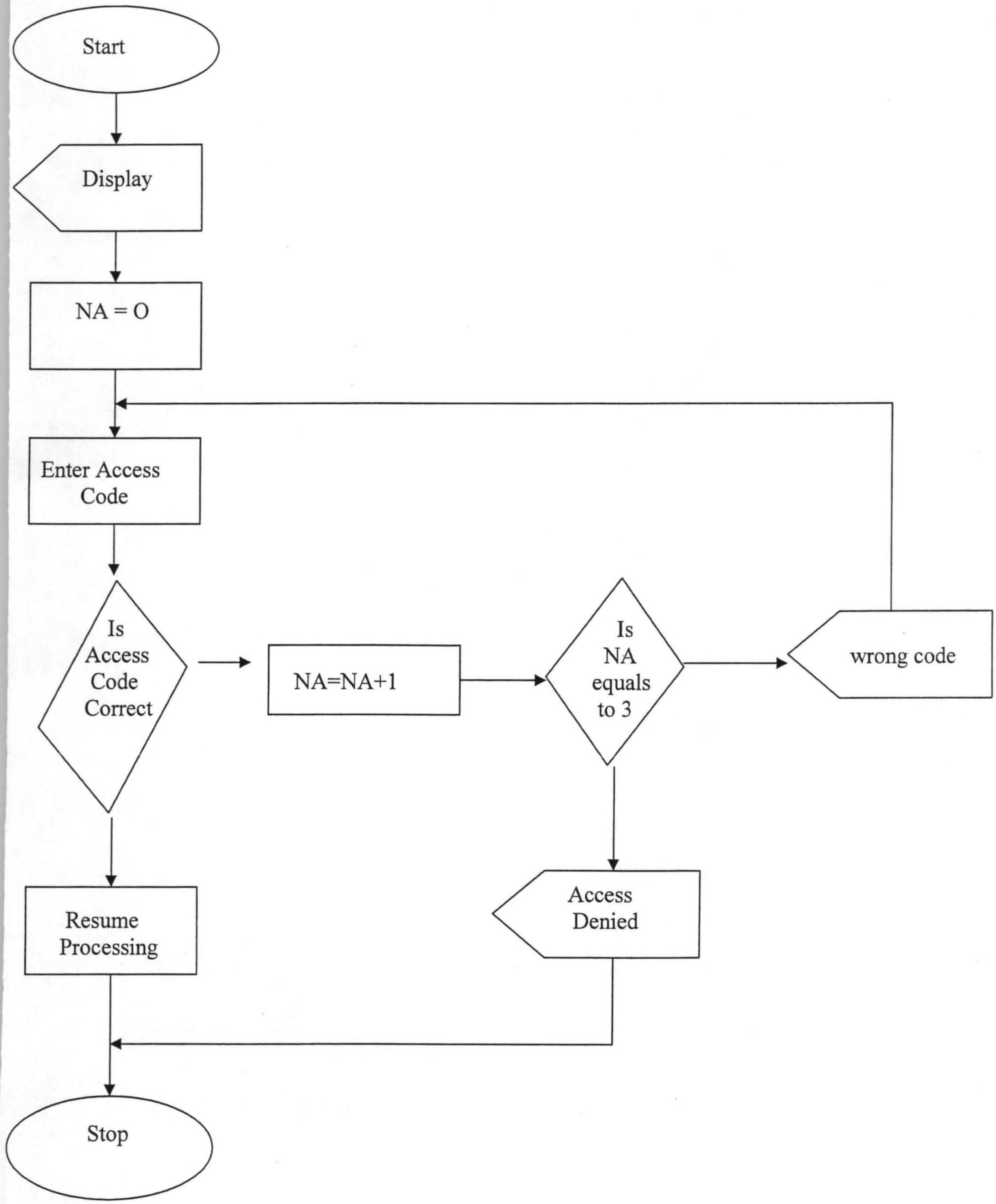
The designed system must not only meet the the requirements of the new system but must also fit into the hardware in which the system will operate. This section outlines the features of the computer system used in the implementation of the software.

Below is therefore the hardware configuration of the computer system used in implementing this research

Project:

(i)RAM------894MB

(ii)DESK JET PRINTER---HP3650



Start
Display
NA = O
Enter Access Code
Is Access Code Correct
NA=NA+1
Is NA equals to 3
wrong code
Access Denied
Resume Processing
Stop

(iii) HARD DISK 64.4GB

(iv) CPU---PIV with 1.80GH3

Software

Below is the software configuration of the computer used in implementing this research work

Windows vista
Microsoft visual basic 6.0

3.4.2 Logical design

In this design, we try to get a blue a blue-print for the new system function. This produces the detail description of the major features of the new system to meet the objectives. This includes the system flowchart as shown below;

3.4.3 Physical design

This phase of the system design aids in the implementation of the users and system requirement as mentioned above. The specifications as follows are essential in this phase.

(a)HARDWARE SPECIFICATION as outlined in 3.4.1

(b)SOFTWARE SPECIFICATION as outlined in 3.4.1

(c)PERSONNEL SPECIFICATION

(d)LANGUAGE TYPE

(e)APPLICATION TYPE

(f)IMPUT PARAMETERS

(g)DOCUMENTATION

(h)PROGRAM MODULES

3.4.3(c) personnel specification

Putman and Myers 1992 advocated that projects should be splitted into small segments and each segment should be handled by a small group of not less than seven person-months of effort, and not less than three-months of effort. This will enhance few communication links, and the work will be done on time and within budget.

3.4.3(d)Language type

Microsoft visual basic 6.0

3.4.3(e) Environment (application type)Web development

The web development project type has an exponential size penalty factor of 1.030 as contained in table 1 below, scheduled multiplier factor of 3.10 as contained in TABLE 2 below, e.t.c

TABLE 1: TYPICAL SIZE PENALTY FACTORS FOR VARIOUS PROJECT TYPES

PROJECT TYPE	EXPONENTIAL SIZE PENALTY FACTOR
CO COMO 2 DEFAULT	1.052
EMBEDDED DEVELOPMENT(BUSINESS)	1.110
E-COMMERCE DEVELOPMENT	1.030
WEB DEVELOPMENT	1.030
MILITARY DEVELOPMENT	1.072

SOURCE: BOEHM ET AL (1995)

Below are the input parameters for the project contained in TABLE 2

TABLE 2: INPUT PARAMETERS FOR THE PROJECT

PARAMETERS	INPUT VALUE
PROCESS PRODUCTIVITY INDEX(PI)	7.00
SIZE PENALTY	1.030
LINEAR PRODUCTIVITY FACTOR	3.30
SCHEDULE MULTIPLIER	3.10

3.4.3(g) Documentation

The parameter s below, TABLE 3 referred, as contained in Boehm (1987) help us to compute the number of pages required to document the specified manuals. The formula used is :

PAGES =A +B *Effort

Where A,B and C are parameters whose values are contained in the empirical deliverables provided by BOEHM (1987).For the purpose of this project, only the software user manual (SUM) number of pages is computed.

DELIVERAVBLES	A	B	C
SOFTWARE USER MANUAL (SUM)	15.00	2.10	0.91

TABLE 3: SAMPLE DELIVERABLES PLAN STANARDS

DELIVERABLES	A	B	C	DESCRIPTION
Software requirements specification(SRS)	3.00	1.21	0.91	Specifies requirements for a computer program and the methods to be used to ensure that each requirement has been met
Software test plan(STP)	5.0	0.12	0.91	Describes the plans for acceptance testing of a program.It describes the software test environments, identifies the tests to be performed and provides schedule for test activities
Software user manual(SUM)	15.0	2.10	0.91	Instructs a hand on software user how to install and use a computer program.It includes a tutorial describing how to accomplish specific user

				work tasks identified as operational scenarios in the software requirements, specification and it also offers a comprehensive reference to all screens, reports and menu choices.
Prototype	1.00	0.00	1.00	System prototype
Functional specification	0.00	4.00	0.91	Defines back-end and middle-tier functional requirements from a business perspective.

SOURCE: (BOEHM,1987)

3.4.4 Program Modules

The under listed estimated modules will constitute the system when fully developed.

(i) Security Module

(ii) Size Modules

(iii) Effort Modules

(iv) Schedule Module

(v) Feasibility Study Module

(vi) Cost Module

Below is the detailed design of each of the modules mentioned above.

3.4.4 (i) Security Module

Welcome the User

Ask for a password

The user can make at most three attempts

The password must not appear on the screen

If the password is correct, call the login menu

3.4.4(ii) Log-in menu

Prompt the user for the project name

Prompt the user for his/her name

Prompt the user for his /her role in the project.

Prompt the user for the date of transaction

Display the menu module

3.4.4(iii) Menu Module

Display the menu header

Display the option to compute size

Display the option to compute Effort

Display the option to compute Schedule

Display the option to compute Cost
Display the option to compute feasibility Time, Effort and Cost

Exit the program

3.4.4(iv) Size Module

Using SLIM modules, Putnam and Myers (1992)

(a) Use Quantitative software Management (QSM) Standard Productivity Index (PI)

For process productivity, Table 4 referred

(b) Estimate the percentage of computed size, Effort and schedule

© Use SLIM software Equation for the computation.

Size = Process Productivity x $\text{effort}^{1/3}$ x $\text{schedule}^{4/3}$

PUTNAM (1978)

Process Productivity = $\text{size}^{B} / (\text{Effort}^{1/3} * \text{Schedule}^{4/3})$

Effort (E) = PI * $\text{Size}^{\text{penalty}}$

Schedule = $(\text{Size}/\text{PI}^3 * \text{Effort})^{0.25}$

Where B is the penalty factor accounting for the relative economies and diseconomies of scale encountered as the project size increases.

If B<1.0 the project size is doubled, the effort is less than double. The project productivity increases as the project size is increased. If B=1.0 the economies and diseconomies of scale are balanced. This linear model is often used for cost estimation of small projects.

If B>0 the project exhibits diseconomies of scale .This is generally due to the following factors

Growth of interpersonal communication overhead

Growth of large system Integration overhead

Banker et al (1994)

Effort α Schedule^3

Manpower buildup = Effort / Schedule^3

3.4.4 (v) Effort Module

Using the computed size in 3.4.3 (iv) above, compute Effort using the formula below

Effort (E1) =productivity *$SLOC^{penalty}$ Where penalty is the exponential size penalty factor (1.030) for web development (Boehm et al,1995) as shown in table 2.

TABLE 4: COMMON VALUES FOR THE LINEAR PRODUCTIVITY FACTORS

PROJECT TYPE	LINEAR PRODUCTIVITY FACTOR
COCOMO 2 DEFAULT	2.94
EMBEDDED DEVELOLPMENT	2.58
E-COMMERCE DEVELOPMENT(BUSINESS)	3.60
WEB DEVELOPMENT	3.30
MILITARY DEVELOPMENT	2.77

SOURCE:BOEHM ET AL (1995)

Send the following parameters to the module: Average size ,penalty factor and productivity factor

TABLE 5: NON LINEAR ENVIRONMENTAL FACTORS

FACTORS	VERY LOW	NOMINAL	EXTRA HIGH
Architecture/Risk Resolution	0.0423	0.014	-0.0284
Development flexibility	0.0223	0.002	-0.0284
Precedentedness	0.0336	0.0088	-0.0284
Process maturity	0.0496	0.0814	-0.0284
Team cohesiveness	0.0264	0.0045	-0.0284

3.4.4 (vi) Schedule Module

Compute the non linear environmental factors by selecting between very low, nominal and extra high for each factor and add up from table 5

Then add the sum to the adjusted penalty factor arrived at the new penalty

Calculate the new Effort 2 using the new penalty, penalty (1)

EFFORT 2(E2) =Productivity $*KSLOC^{penalty1}$

Find the cube root of the newly computed Effort obtained from the preceding computation.

Multiply the result by the schedule multiplier 3.10, for web development taken from table 6 below:

Compute the schedule using the formula below:

SCHEDULE = 3.10 * (Effort 2) $^{0.33}$

Feasibility study module 3.4.4 (vii)

Use the computed Total Effort and Schedule and then compute the time and Effort needed to carry out feasibility study using the following:

EFFORT 5% to 10% Total EFFORT

(10% of the Total Effort was used in the computation)

SCHEDULE 25% of the Total Schedule

3.4.4 (Viii) Cost Module

Cost = Effort * Labor rate of the organization

CHAPTER FOUR

SYSTEM CONSTRUCTION

4.0 Introduction

In this chapter discussion will be based on choice of programming language, testing of program and debugging of program, system implementation and finally system documentation.

4.1 Choice of Programming Language

The importance of the type of programming language chosen for the development of any software project cannot be over-emphasized. As a result of the above reason, certain criteria must be considered before a particular programming language is chosen. These criteria are as follows:

(i) Availability of the language

There are some languages that are dependant on the architecture of the computer system. As a result, they have restricted applicability. However, preference is given to languages that can easily be transferred from one computer to another .Hence, languages with wide bases of applicability and user acceptance are preferred.

(ii) Familiarity with the language

Another issue is familiarity with the language. This has had entirely too much influence in language selection. All else being equal, it is sensible to use a programming language where there is a large market of relatively skilled programmers familiar with it.

(iii) The Language of Choice

Based on the various criteria used for choice of programming language, the language of choice in this project is Microsoft Visual Basic 6.0.

A strong practical support for Visual Basic is the fact that it is nearly universally accepted. Hence it is "good enough" for many student projects. Visual Basic is easy to learn, which makes it an excellent tool for understanding elementary programming concepts. In addition, it has evolved into such a powerful and popular product that skilled programmers are in demand in the job market.

As the world turns to graphical user interfaces, computer programming languages are changing to accommodate the shift. Visual Basic 6.0 is designed to allow the programmer to develop applications that run under windows without the complexity generally associated with windows programming.

4.2 Software Testing

Software testing is the process of testing the functionality and correctness of software by running it. The aim of the software testing is to dictate errors before the software is delivered. Another purpose of conducting software testing is for reliability estimation. The problem of applying software testing can only suggest the presence of errors or flaws not their absence (unless the testing is exhaustive).

The problem of applying software testing to reliability estimation is that the input distribution used for selecting test case may be flawed.

However, the program in this project was written in modules and these modules were tested independently. After independent testing of the modules an integrated test was carried out on the module to ensure that they were logically interwoven.

4.3 System Implementation

This involves the effort of the user department in setting the system into operation. And its aim to translate the plans, schedules and designs from the feasibility study into integrated functional operations. There are three basic implementation methods adopted by system developers. These include feasibility study into integrated functional operations. There are three basic implementation methods adopted by system developers. These include parallel run methods, direct change over method as well as pilot run method. The method used in this project is direct change over.

Direct Change Over Method: This method involves the direct change over from the old system to the new system having realized the new system can operate alone as a complete system. Since it is capable of achieving the purpose for which it was designed.

4.4 System Documentation

System documentation is an act of keeping track of work produced during software cost estimation by collecting, organizing, sorting and also maintaining on paper, a complete record of the activities carried out in the development of the application and procedure on how it is to be used.

The following are the reasons for documentation:

(i) It enabled us to handle complexity by providing means of referencing the component of the model.
(ii) It facilitates communication during development between the team members and also between the users and the developers after the development of the system.
(iii) It assists in maintenance through the use of comments which explain the logic of the program to the maintenance programmer, who would be required to modify the program in future if the need arises especially when the developer is not within reach. So appropriate documentation was carried out from the survey phase through all the necessary phase from systems development to training exercise. Its presence no doubt will enhance an efficient service from the package.

CHAPTER FIVE
SUMMARY, RECOMMENDATIONS AND CONCLUSION

5.0 Introduction

This chapter closes the research work with general recommendation, summary and finally conclusion. After this the references of the materials consulted during this research work is then listed.

5.2 Summary

Cost Estimation in software development is not easy task However, in this research work a software tool has been developed, which when properly used will improve the prediction or estimate of the cost of a project.

In this research, various methods of estimating cost of software development were analyzed with their various advantages and disadvantages. However, it had been discovered that the developed tool will provide a better estimate than its predecessors.

Cost estimation tool when used by project managers will help them to give proper cost of the project to their clients.

This tool will be helpful to either an analyst or programmer to aid him or her in the system and design job shown in the test case (case study).

5.2 Recommendation

Sequel to the achieved objectives of this research work, it is necessary to make obvious suggestions as to what measures to adopt to enable the new system stand the test of time. These measures include the following recommendation for this project:

(i) Analyze the metrics at the project and apply the lessons to future projects.

The organization should ensure that data is properly documented for future reference.

(ii) The project manager should ensure that everyone in the team is educated on how to use the tool.

5.3 Conclusion.

Software Cost estimation is simple in concept but difficult and complex in reality.
The difficulty and complexity required for estimates exceed the capabilities of most software project managers. As a result, manual estimates are not sufficient for large applications above roughly 1000 function points in size.

Commercial software cost estimation tools can often outperform manual human estimates in terms of speed and cost effectiveness. However, no method of estimation is totally error free. The current best practice for software cost estimation is to use a combination of software cost estimation tools coupled with software project management tools, under the careful guidance of experienced software project managers and estimation specialists.

REFERENCES

1 Roetzheim, William H and Beasley, Reyna A; Best Practices in Software Cost and Schedule Estimation; Prentice Hall PTR Saddle River, NJ;1998

2 Kemerer, Chris F; "An Empirical Validation of Software Cost Estimation Models; Communications of the ACM; 30 May 1987;

3 Jones , Capers; "The Costs, Schedule, and value of Software Process Improvement"; SPR Technical Report; Software Productivity Research, Inc; Burlington, MA, January 1998

4 Boehm, Barry W et al, Software Cost Estimation with COCOMO II, prentice Hall, 2000

5 Kim Johnson, 2005. Software Cost Estimation : Metrics and Models. Department of Computer Science University of Calgary Alberta, Canade.

6 Oyong, Samuel Bisong (2003): Time Estimation Tool for Software Development Projects .Abubakar Tafawa Balewa University, Bauchi.

7 Ikwun, Ushie G. (2001): Introduction to System Analysis and Design, University of Calabar Computer Centre.

8 Hareton, L. Z. F, 2002 Software Cost Estimation. Department of Computing, The Hong Kong Polytechnic University.

9 Programming in Visual Basic. Version 6.0.Julia Case Bradley and Anita C. Millspaugh (2002).

Login Module

```
Option Explicit
Dim trialNum As Integer
Private Sub cmdLogin_Click()
trialNum = trialNum + 1
With dc1
  If txtUsername.Text = .Recordset.Fields(0).Value And txtpassword.Text =
.Recordset.Fields(1).Value Then
     Me.Hide
     frmmain.Show
  Else
     If trialNum > 3 Then
        MsgBox "Login trial exceeded. Login Failed!"
        End
     Else
        MsgBox "Invalid Username/Password"
     End If
  End If
End With
End Sub

Private Sub Form_Load()
trialNum = 1

dc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source = " &
App.Path & "\bank.mdb"
dc1.RecordSource = "settings"
dc1.Refresh
End Sub

Private Sub Form_Unload(Cancel As Integer)
End
End Sub

Private Sub lbxit_Click()
End
End Sub
```

Main Module

```
Public Sub loadCurrentProject(projectName As String)
  If projectName <> "" Then
     lbprojectname.Caption = projectName
     btnSize.Enabled = True
     btnEffort.Enabled = True
```

```
        btnSchedule.Enabled = True
        btnCost.Enabled = True
        cmdOutput.Enabled = True
        'btnFeasibilty.Enabled = True
        'btnArch.Enabled = True
    End If
End Sub

Private Sub btnCost_Click()
frmCost.Show
End Sub

Private Sub btnEffort_Click()
frmEffort.Show
End Sub

Private Sub btnExit_Click()
End
End Sub

Private Sub btnSchedule_Click()
frmSchedule.Show
End Sub

Private Sub btnSize_Click()
frmSize.Show
End Sub

Private Sub cmdOutput_Click()
frmReport.Show vbModal
End Sub

Private Sub Label4_Click()
frmNewProject.Show vbModal
End Sub

Private Sub MDIForm_Load()
Timer1.Enabled = True
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
End
End Sub

Private Sub Timer1_Timer()
Timer1.Enabled = False
```

```
dc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source = " &
App.Path & "\bank.mdb"
dc1.RecordSource = "settings"
dc1.Refresh

btnSize.Enabled = False
btnEffort.Enabled = False
btnSchedule.Enabled = False
btnCost.Enabled = False
cmdOutput.Enabled = False
'btnFeasibilty.Enabled = False
'btnArch.Enabled = False

If dc1.Recordset.Fields(2).Value = "yes" Then
   frmStartup.Show vbModal
End If

End Sub
```

Size Module

```
Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdCompute_Click()
Dim productivity As Double
Dim effort As Double
Dim schedule As Double
Dim pSize As Double

productivity = Val(txtproductivity.Text)
effort = Val(txtEffort.Text)
schedule = Val(txtSchedule.Text)
pSize = Val(txtSize.Text)

pSize = productivity * (effort ^ (1 / 3)) * (schedule ^ (4 / 3))
txtSize.Text = pSize
End Sub

Private Sub cmdSave_Click()
```

```
With dc1
   .Recordset.Update
   .Recordset.Fields(1) = Date
   .Recordset.Fields(3) = txtproductivity.Text
   .Recordset.Fields(4) = txtEffort.Text
   .Recordset.Fields(5) = txtSchedule.Text
   .Recordset.Fields(2) = txtSize.Text
   .Recordset.Update
End With
End Sub

Private Sub Form_Load()
dc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source = " &
App.Path & "\bank.mdb"
dc1.RecordSource = "SELECT * FROM computation WHERE project_name = '" &
frmmain.lbprojectname.Caption & "'"
dc1.Refresh

With dc1
   Me.Caption = .Recordset.Fields(0).Value & " SIZE (Modified: " &
.Recordset.Fields(1).Value & ")"
   txtproductivity.Text = .Recordset.Fields(3).Value
   txtEffort.Text = .Recordset.Fields(4).Value
   txtSchedule.Text = .Recordset.Fields(5).Value
   txtSize.Text = .Recordset.Fields(2).Value
End With

End Sub
```

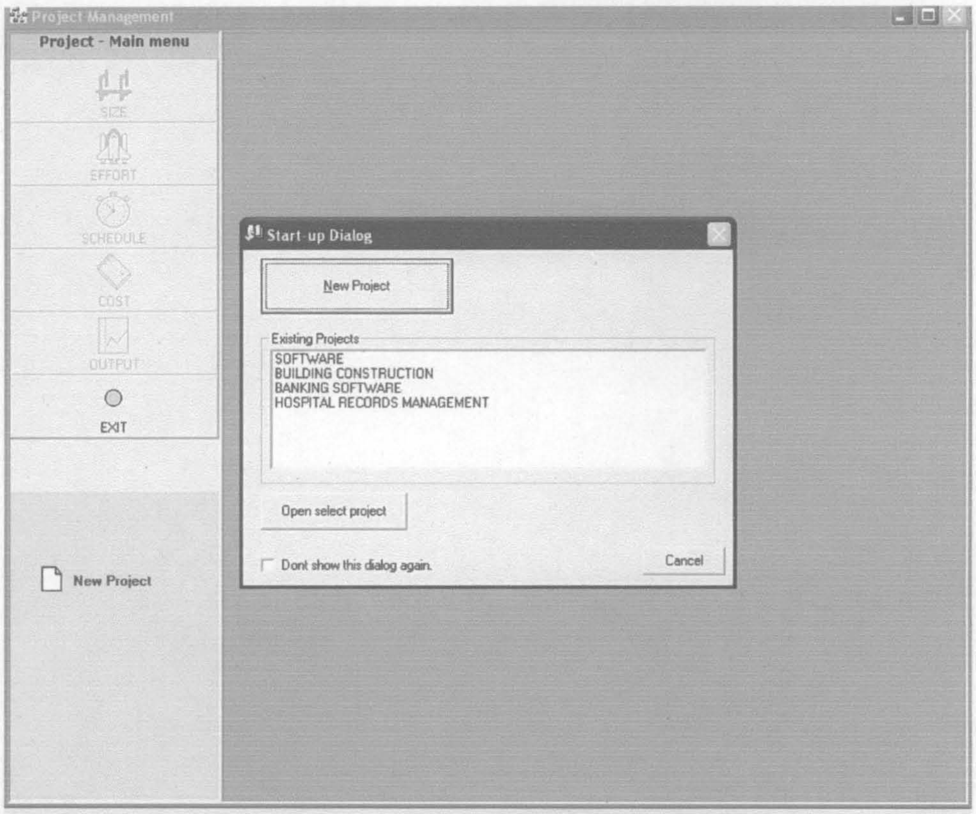
Effort Module

```
Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdCompute_Click()
Dim effort As Double
Dim pi As Double
Dim pSize As Double
Dim penalty As Double

pi = 3.142
pSize = Val(txtSize.Text)
penalty = Val(txtPenalty.Text)

effort = pi * (pSize ^ penalty)
```

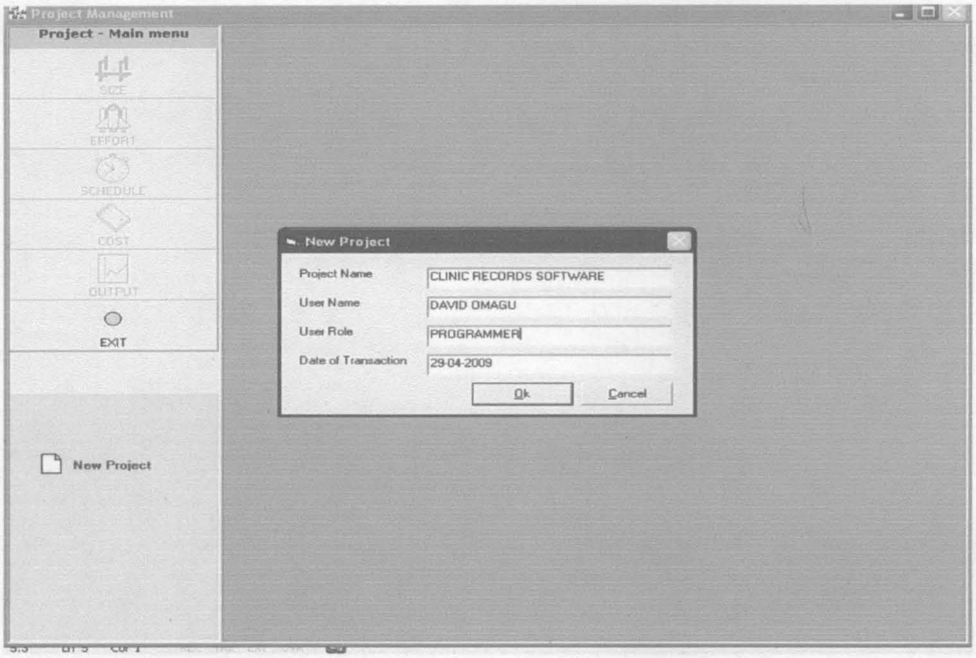
```
txtEffort.Text = effort

End Sub

Private Sub cmdSave_Click()
With dc1
   .Recordset.Update
   .Recordset.Fields(1) = Date
   .Recordset.Fields(4) = txtEffort.Text
   .Recordset.Fields(2) = txtSize.Text
   .Recordset.Fields(6) = txtPenalty.Text
   .Recordset.Update
End With
End Sub

Private Sub Form_Load()
dc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source = " &
App.Path & "\bank.mdb"
dc1.RecordSource = "SELECT * FROM computation WHERE project_name = '" &
frmmain.lbprojectname.Caption & "'"
dc1.Refresh

With dc1
   Me.Caption = .Recordset.Fields(0).Value & " EFFORT (Modified: " &
.Recordset.Fields(1).Value & ")"
   txtEffort.Text = .Recordset.Fields(4).Value
   txtSize.Text = .Recordset.Fields(2).Value
   txtPenalty.Text = .Recordset.Fields(6).Value
End With

End Sub
```

Schedule Module

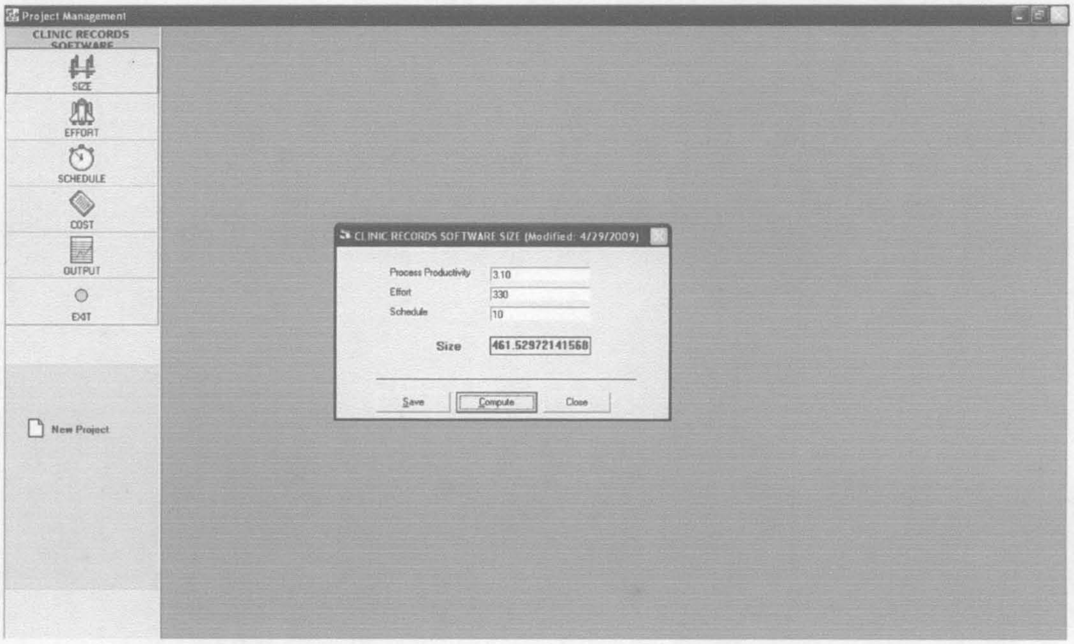
```
Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdCompute_Click()
Dim schedule As Double
Dim pSize As Double
Dim effort As Double
Dim pi As Double

pSize = Val(txtSize.Text)
effort = Val(txtEffort.Text)
```



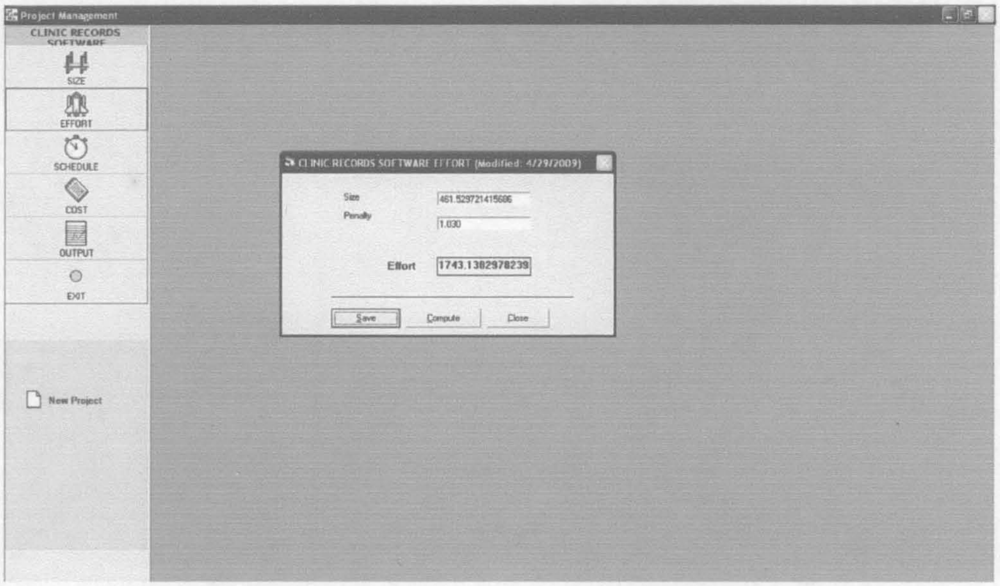
START UP DIALOG



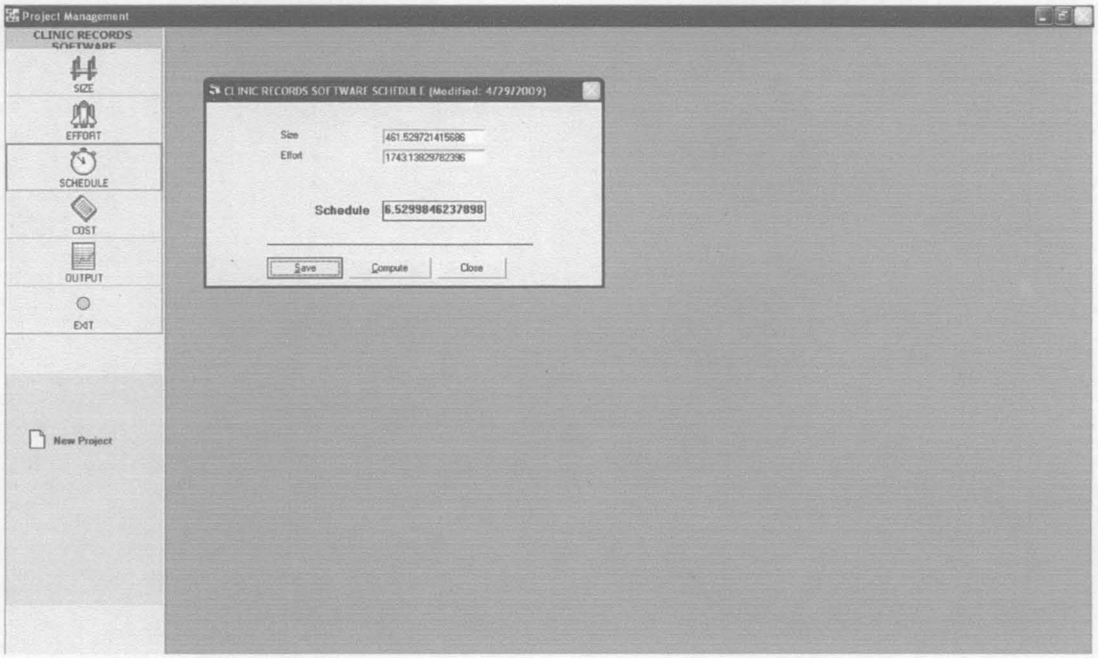
NEW PROJECT INTERFACE


Project Management
CLINIC RECORDS SOFTWARE
SIZE
EFFORT
SCHEDULE
COST
OUTPUT
EXIT
New Project
CLINIC RECORDS SOFTWARE SIZE (Modified: 4/29/2009)
Process Productivity
3.10
Effort
330
Schedule
10
Size
461.52972141568
Save
Compute
Close

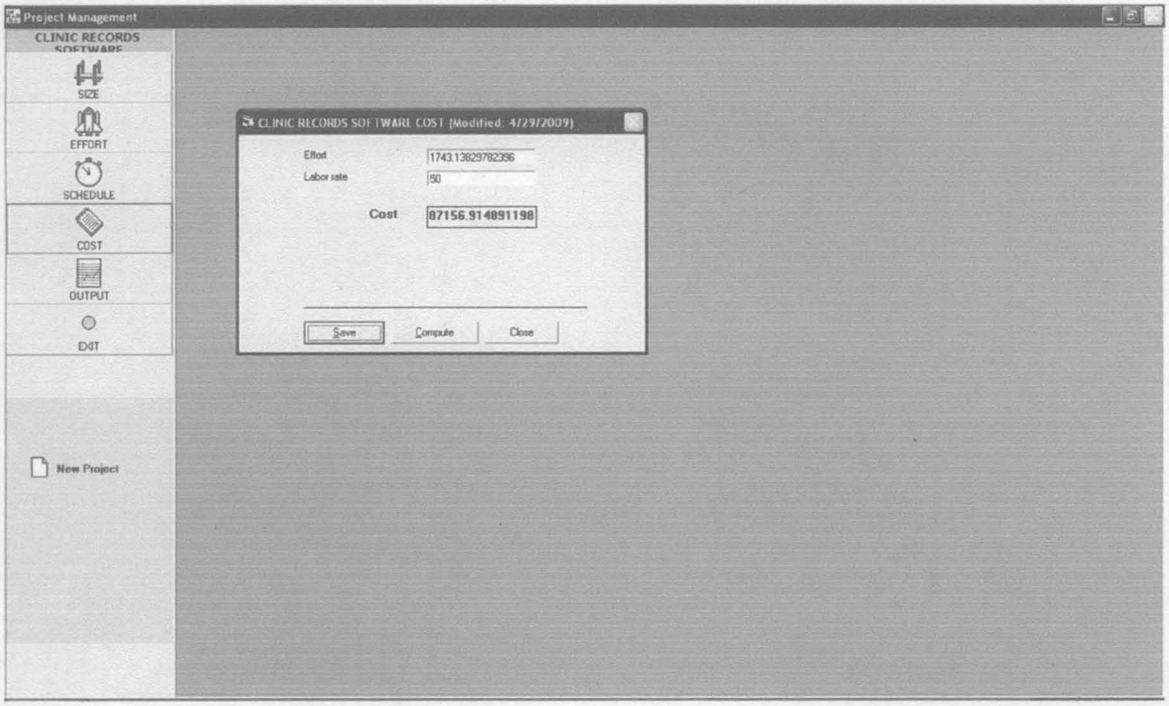
SIZE MODULE



EFFORT MODULE



SCHEDULE MODULE


Project Management
CLINIC RECORDS SOFTWARE
SIZE
EFFORT
SCHEDULE
COST
OUTPUT
EXIT
New Project
CLINIC RECORDS SOFTWARE COST (Modified: 4/29/2009)
Effort
1743.13829782396
Labor rate
50
Cost
87156.914891198
Save
Compute
Close

COST MODULE

HOSPITAL RECORDS MANAGEMENT

PROJECT MANAGEMENT

Date: 4/30

Main

Size	1157.11291753393
Effort	332
Schedule	10.8
Cost	0

Miscellaneous

Process Productivity	7
Penalty	0
Labour Rate	0

Prepared by: