K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent college of Somaiya Vidyavihar University)

Batch: A2 Roll No.: 16010121045

Experiment / assignment / tutorial No.____

TITLE: Project Metric estimations for Mini Project

AIM: To enable the students learn different techniques for performing software size and cost estimation

Expected Course outcome of Experiment:

CO 1: Understand the software development process and Estimate different types of resources for the given project.

Books/ Journals/ Websites referred:

1. Roger Pressman, “Software Engineering”, sixth edition, Tata McGraw Hill.
2. http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html
3. http://sunset.usc.edu/research/COCOMOII/expert_cocomo/expert_cocomo2000.html
   http://softwarecost.org/tools/COCOMO/

Pre Lab/ Prior Concepts:

Software projects have tendency of going past their deadline, going over budget, or both. The problem lies in the estimation of the amount of effort required for the development of a project. The cost estimation is usually dependent upon the size estimate of the project, which may use lines of code or function points as metrics. There are several different techniques for performing software cost estimation, including expert judgement and algorithmic models. Estimation by expert judgement is a common way of estimating the effort required for a project. Unfortunately, this method of estimation does not emphasize re-estimation during the project life cycle, which is an important part of project tracking, because it allows the estimates to be improved during the project life cycle. The quality of a cost estimation model is not so much attributed to the initial estimate, but rather the speed at which the estimates converges to the actual cost of the project. COCOMO is a popular algorithmic model for cost estimation whose cost factors can be tailored to the individual development environment, which is important for the accuracy of the cost estimates. More than one method of cost estimation should be done so that there is some comparison available for the estimates. This is especially important for unique projects. Cost estimation must be done more diligently throughout the project life cycle so that in the future there are fewer surprises and unforeseen delays in the release of a product.

Estimation of size and cost of the developing project is required for the following major decision situations

- Financial decisions involving a software development effort.
- Setting project budgets and schedules as a basis for planning and control.
- Deciding on or negotiating tradeoffs among software cost, schedule, functionality, performance or quality factors.
- Making software cost and schedule risk management decisions.
- Deciding which parts of a software system to develop, reuse, lease, or purchase.
- Making legacy software inventory decisions: what parts to modify, phase out, outsource, etc
- Deciding how to implement a process improvement strategy, such as that provided in the SEI CMM

Defining Cost estimation:

Cost estimation can be defined as the approximate judgement of the costs for a project. Cost estimation will never be an exact science because there are too many variables involved in the calculation for a cost estimate, such as human, technical, environmental, and political. Furthermore, any process that involves a significant human factor can never be exact because humans are far too complex to be entirely predictable. Furthermore, software development for any fair-sized project will inevitably include a number of tasks that have complexities that are difficult to judge because of the complexity of software systems.

Cost estimation is usually measured in terms of effort. The most common metric used is person months or years (or man months or years). The effort is the amount of time for one person to work for a certain period of time. It is important that the specific characteristics of the development environment are taking into account when comparing the effort of two or more projects because no two development environments are the same. A clear example of differences in development environments are the amount of time people work in different countries; the typical workweek in North America is 40 hours per week, while in Europe the typical workweek is 35 hours per week. Thus, when comparing a project from North America with a project from Europe, a conversion factor would have to be used to all for an accurate comparison. Different variables can be used for cost estimation, which leads to a difficulty when comparing projects if standard models or tools are not used. For example, a cost estimate can include factors from management, development (e.g., training, quality assurance), and other areas specific to an organization.

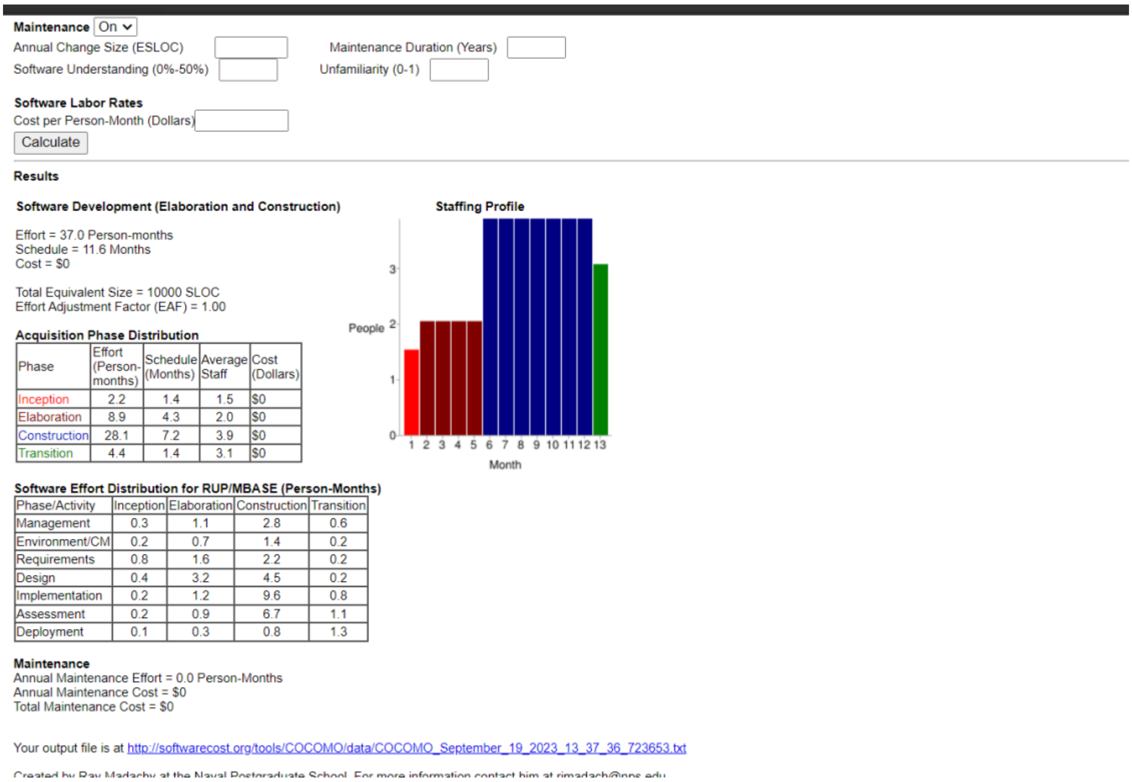
Estimator:

The people who do the cost estimates could be either directly or indirectly responsible for the implementation for a project, such as a developer or manager, respectively. Someone who has knowledge of the organization and previous projects could use an analogy-based approach to compare the current project with previous projects, which is a common method of estimation for small organizations and small projects. The historical data is often limited to the memory of the estimator. In this case, the estimator would need to be experienced and would likely have been with the company for awhile.

Some people believe it is better if the estimates are done by outsiders so that there is less chance of bias. It is true that people outside an organization will likely have to deal with fewer company politics than people within the organization. For example, the developer for a company may want to please the manager and so give an estimate that is overly-optimistic. The disadvantage of having an outside estimate is that the person would have less knowledge of the development environment, especially if the person is from outside the company. An empirical method of estimation would then be required, such as the Constructive Cost Model (COCOMO). Empirical methods of estimation can be used by all types of estimators. There may be some resistance to using an empirical method of estimation because there may be some question on whether a model could outperform an expert. People who are accurate estimators are rare in our experience, and so it is best to get the opinion of several people or tools.

Cost estimation using different COCOMO models:

(Attach screen shots of the size/cost estimations using different COCOMO tools. Justify estimation details



Maintenance On

Annual Change Size (ESLOC)

Maintenance Duration (Years)

Software Understanding (0%-50%)

Unfamiliarity (0-1)

Software Labor Rates

Cost per Person-Month (Dollars)

Results

Software Development (Elaboration and Construction)

Effort = 37.0 Person-months
Schedule = 11.6 Months
Cost = $0

Total Equivalent Size = 10000 SLOC
Effort Adjustment Factor (EAF) = 1.00

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2.2	1.4	1.5	$0
Elaboration	8.9	4.3	2.0	$0
Construction	28.1	7.2	3.9	$0
Transition	4.4	1.4	3.1	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.3	1.1	2.8	0.6
Environment/CM	0.2	0.7	1.4	0.2
Requirements	0.8	1.6	2.2	0.2
Design	0.4	3.2	4.5	0.2
Implementation	0.2	1.2	9.6	0.8
Assessment	0.2	0.9	6.7	1.1
Deployment	0.1	0.3	0.8	1.3

Maintenance
Annual Maintenance Effort = 0.0 Person-Months
Annual Maintenance Cost = $0
Total Maintenance Cost = $0

Your output file is at http://softwarecost.org/tools/COCOMO/data/COCOMO_September_19_2023_13_37_36_723653.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Conclusion:

Cost estimation using cocomo model was done successfully.

Post Lab Descriptive Questions

1. **You are appointed as an estimator to find out efforts required to implement following project. The goal of the project is to create a database of all Hindi films released since 2000. The software would allow one to generate a list of top ten hit films, top ten flop films, best comedy films, and so on. Using your prior experience you have decided the approximate sizes of each module of the software as follows:**
   Data entry (0.9 KDSI)
   Data update (0.7 KDSI)
   Query (0.9 KDSI)
   Report generation and display (2 KDSI)

   Also take into consideration the following cost drivers with their ratings:

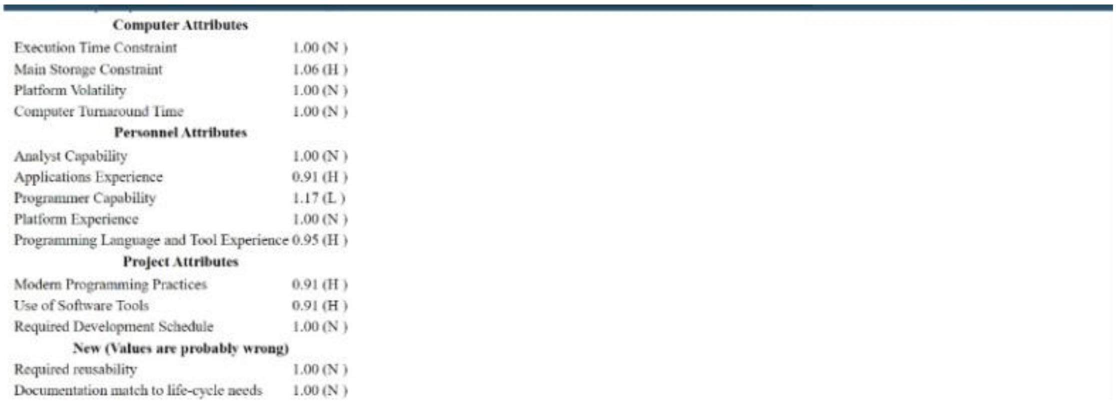
Storage constraints (Medium)
Experience in developing similar software (High)
Programming capabilities of the developers (Low)
Application of software engineering methods (High)
Use of software tools (Medium)
(All other cost drivers have nominal rating).

Now answer the following:

Applying intermediate COCOMO estimate the minimum size of the team you would require to develop this system
Assuming that your client would pay Rs. 50,000 per month of development, how much would be the likely billing?

a) **Identify type of a project as per COCOMO**
b) **Prepare an estimate of required effort and cost**

Limitations: Values presented here are arbitrary and doesn't relate to real life



Computer Attributes	
Execution Time Constraint	1.00 (N)
Main Storage Constraint	1.06 (H)
Platform Volatility	1.00 (N)
Computer Turnaround Time	1.00 (N)
Personnel Attributes	
Analyst Capability	1.00 (N)
Applications Experience	0.91 (H)
Programmer Capability	1.17 (L)
Platform Experience	1.00 (N)
Programming Language and Tool Experience	0.95 (H)
Project Attributes	
Modern Programming Practices	0.91 (H)
Use of Software Tools	0.91 (H)
Required Development Schedule	1.00 (N)
New (Values are probably wrong)	
Required reusability	1.00 (N)
Documentation match to life-cycle needs	1.00 (N)

2. Explain COCOMO II model.

COCOMO-II is the revised version of the original Cocomo (Constructive Cost Model). It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity. It consists of three sub-models:

1. End User Programming:
Application generators are used in this sub-model. The end user writes the code by using these application generators.
Example – Spreadsheets, report generator, etc.

2. Intermediate Sector:
(a). Application Generators and Composition Aids –

This category will create largely prepackaged capabilities for user programming. Their product will have many reusable components. Typical firms operating in this sector are Microsoft, Lotus, Oracle, IBM, Borland, Novell.

(b). Application Composition Sector –

This category is too diversified and to be handled by prepackaged solutions. It includes GUI, Databases, domain specific components such as financial, medical or industrial process control packages.

(c). System Integration –

This category deals with large scale and highly embedded systems.

3. Infrastructure Sector:

This category provides infrastructure for the software development like Operating System, Database Management System, User Interface Management System, Networking System, etc.

Stages of COCOMO II:

Stage-I:

It supports estimation of prototyping. For this it uses Application Composition Estimation Model. This model is used for the prototyping stage of application generator and system integration.

Stage-II:

It supports estimation in the early design stage of the project, when we know less about it. For this it uses Early Design Estimation Model. This model is used in the early design stage of application generators, infrastructure, system integration.

Stage-III:

It supports estimation in the post architecture stage of a project. For this it uses the Post Architecture Estimation Model. This model is used after the completion of the detailed architecture of application generator, infrastructure, system integration.