International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

COCOMO and COCOMO II Model- A Case Study

Prof. Ronica Raj[1], Prof. Rukhsar Haji[2], Farheen Rizvi[3], Naushin Khan[4]

[1]*Professor, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India*
[2]*Professor, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India*
[3]*Student, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India*
[4]*Student, Dept. of Computer Engineering, Rizvi College of Engineering, Maharashtra, India*

---***---

Abstract – *One of the main challenges for software, nowadays, is software cost estimation. It implies estimating the cost of all activities including software development, design, supervision, maintenance and so on. Finding precise cost-estimation of software projects optimizes the internal and external processes, staff works, efforts and the overheads to be coordinated with one another. The COCOMO Model is well known as the currently predominate model for software cost estimation. It allows one to work from linguistic variables to as far as estimating software project effort and schedule.*

***Key Words*: COCOMO model, COCOMO II, cost estimation, software engineering.**

1. INTRODUCTION

The COCOMO (Constructive Cost Model) is an empirical model that was derived with the help of gathering data from a large number of software projects. These data were analyzed to determine formulae that are the best fit to the observations. These formulae associate the size of the system and product, project and team factors to the effort to develop the system.

COCOMO 81, first version of the COCOMO model was a three-level model. COCOMO 81 thinks that the software would be developed according to a waterfall process using standard programming languages like C.

To take these modifications into account, the COCOMO II model identifies different approaches to software development like prototyping. COCOMO II helps in development of a spiral model and embeds number of sub-models that create the whole estimates.

Motivation: One of the most difficult phases in the software development process is the ability to give accurate time estimations for a project. Becoming a software company, is an increasingly-frequent goal in organizations varying from finance, transportation, aerospace, electronics, to manufacturing firms. Competitive advantage is highly dependent on the development of smart, tailorable products and services, and on the ability to develop and adapt these products and services more rapidly than competitors' adaptation times.

Drastic downfall of computer hardware platform costs and the prevalence of commodity software solutions have indirectly ten to decline systems' development costs. Some of the software cost models that exist, have initiatives addressing aspects of these issues. These new approaches have not been strongly matched hitherto by complementary new models for estimating software costs and schedules. This makes it difficult for organizations in performing effective planning, analysis, and control of projects using the new approaches

2. COCOMO I

The Constructive Cost Model was initially developed by Barry W. Boehm. The model is for estimating effort, cost, and schedule for software projects. It is also called as Basic COCOMO. This model is used to give an approximate estimate of the various parameters of the project. Example of projects based on this model is business system, payroll management system and inventory management systems. COCOMO I is useful in the waterfall models of the software development cycle.

2.1. Types of Models

COCOMO has a hierarchy of three detailed and accurate forms in an increasing order. Using any of the three forms is valid as per requirements. These are types of COCOMO model:

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

The first level, Basic COCOMO can be used for quick and somewhat rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations.

Intermediate COCOMO takes these cost drivers under consideration and Detailed COCOMO additionally accounts for the influence of individual project phases, i.e. in case of Detailed it accounts for both these cost drivers and also calculations are performed phase wise henceforth giving rise to a more meticulous outcome.

It was found that effort is the main cost driver for software development, where effort is translated into cost. The chief constituent which affects the effort estimation is the developed kilo line of code (KLOC). The KLOC include all program instructions and formal statements. Many software cost estimation models where proposed to help in providing a top quality estimate to assist project manager in establishing accurate decision about their projects.

1) Basic Model:

$$E = a(KLOC)^b$$

The given formula is defines the cost estimation of for the basic COCOMO model, and also is used in the subsequent models. The constant values 'a' and 'b' for the Basic Model are:

TABLE I

Software projects	A	B
Organic	2.4	1.05
Semi-Detached	3.0	1.12
Embedded	3.6	1.20

2) Intermediate Model:

The basic Cocomo model assumes that the effort is merely a function of the number of lines of code and some constants evaluated according to the different software system. However, in reality, system's effort and schedule cannot be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, capability are essential. These facets are known as Cost Drivers and the Intermediate Model makes effective use of 15 such drivers for cost estimation. The Intermediate COCOMO formula now takes the form:

$$E = (a(KLOC)^b) * EAF$$

TABLE II

Software projects	A	B
Organic	3.2	1.05
Semi-Detached	3.0	1.12
Embedded	2.8	1.20

3) Detailed Model:

Detailed COCOMO encompasses every attribute of the Intermediate version with an assessment of the cost driver's impact on each step of the software engineering process. The Detailed model uses a variety of effort multipliers for each cost driver attribute. In Detailed COCOMO, the whole software is divided in several modules and then we apply COCOMO in different modules to estimate effort and then sum the effort. The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System design
3. Detailed design
4. Module code and test
5. Integration and test
6. Cost Constructive model

3. COCOMO II

The COCOMO-II is the revised version of the original Cocomo (Constructive Cost Model) and is developed at the University of Southern California. This model calculates the development time and effort taken as the total of the estimates of all the individual subsystems. In this model, whole software is divided into different modules. Example of projects based on this model is Spreadsheets and report generator. COCOMO II helps in development of spiral model and embeds number of sub-models that create whole estimation.

In [2], the COCOMO 2.0 model uses function points and/or source lines of code as the basis for measuring size for the Early Design and Post-Architecture estimation models. For comparable size quantification across COCOMO 2.0 participants and users, standard counting rules are mandatory. A consistent definition for size within projects is a prerequisite for project planning and control, and a harmonious definition across projects is a prerequisite for process improvement.

Fig a shows COCOMO II sub-models as well as they is used.

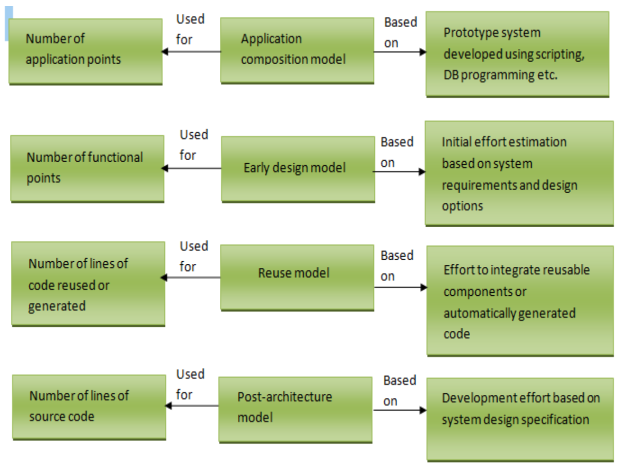
Number of application points
Used for
Application composition model
Based on
Prototype system developed using scripting, DB programming etc.
Number of functional points
Used for
Early design model
Based on
Initial effort estimation based on system requirements and design options
Number of lines of code reused or generated
Used for
Reuse model
Based on
Effort to integrate reusable components or automatically generated code
Number of lines of source code
Used for
Post-architecture model
Based on
Development effort based on system design specification

Fig a. The COCOMO II model

COCOMO II incorporates a variety of sub-models that generate increasingly detailed software estimates .The sub-models in COCOMO II are:

a) Application composition model: Used when software is composed from existing parts.
b) Early design model: Used when requirements are available but design has not yet started.

c) Reuse model: Used to compute the effort of integrating reusable components.
d) Post architecture model: Used once the system architecture has been designed and more information about the system is available.

In [5], COCOMO II estimates utilizes definitions of labor categories, thus they include project managers and program librarians, but exclude computer center operators, personnel-department personnel, secretaries, higher management, janitors, etc. COCOMO II expresses size in thousands of SLOC (KSLOC) and avoids non-delivered support software such as test drivers. They are included in account that they be implemented in the same fashion as distributed code.

4. COMPARISION

COCOMO I	COCOMO II
COCOMO I is convenient in the waterfall models of the software development cycle.	COCOMO II is helpful in non-sequential, rapid development and reuse models of software.
Effort equation's exponent is determined by 3 development modes.	Effort equation's exponent is determined by 5 scale factors.
This model is based upon the linear reuse formula.	This model is based upon the non-linear reuse formula
Development begins with the requirements assigned to the software.	It follows a spiral type of development.

5. CONCLUSIONS

COCOMO II has proven to be an invigorating, flexible, and precise model to utilize in cost estimations therein it urges the user to be creative but at the identical time also responsible. Considering that one effort multiplier single-handedly can impact the hassle product with the assigned factor e.g. if the factor is as low as half some extent, then the hassle is consequently halved. COCOMO II aids that it copes with the rapid change that's seen within the software field today. COCOMO II inputs existing objectives for the system under development in terms of the desires functions, performance, quality and also the environment that's visiting be utilized. Software cost estimation could be a crucial a component of the software development process. The COCOMO suite (COCOMO II model and its extensions) offers a robust instrument to predict software costs.

Unfortunately not all of the extensions are already calibrated and so still experimental. Only the Post-Architecture model is implemented during a calibrated software tool. Despite this disadvantage, the COCOMO II suite helps in managing software projects. It aids in process improvement analyses, tool purchases, architecture changes, component make/buy tradeoffs and deciding process with credible results. A lot of endeavours were done in order to measure up to the conversions in software life cycles, technologies, components, tools, notations and organizational cultures since the first version of COCOMO (COCOMOI, COCOMO 81).

REFERENCES

[1] "A DECISION SUPPORT SYSTEM FOR ESTIMATING COST OF SOFTWARE PROJECTS USING A HYBRID OF MULTI-LAYER ARTIFICIAL NEURAL NETWORK AND DECISION TREE",JavadPashaeiBarbin and Hassan Rashidi ,International Journal in Foundations of Computer Science & Technology (IJFCST) Vol.5, No.6, November 2015
[2] "A review paper on COCOMO model" Gajender pal Manish kumarand KuldeepbaralaDce, ggn,Volume-1 | Issue-4 | April,2015 | Paper-14
[3] https://www.geeksforgeeks.org
[4] "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0*",Barry Boehm, Bradford Clark, Ellis Horowitz, Chris WestlandUSC Center for Software Engineering, Ray MadachyUSC Center for Software Engineering and Litton Data System and Richard SelbyUC Irvine and Amadeus Software Research
[5]DarkoMilicic, "Applying COCOMO- Case study", Department of Software Engineering, School of Engineering Blekinge Institute of Technology Box 520 SE – 372 25 Ronneby Sweden
[6]"An Analysis of Research in Software Engineering: Assessment and Trends",Yutao Ma, School of Computer, Wuhan University, Wuhan 430072, China , Research Center for Complex Network, Wuhan University, Wuhan 430072, China
[7]P.C. Pendharkar, "Probabilistic Estimation of Software Size and Effort", Expert Systems with Applications, Vol. 37, Issue 6, pp. 4435-4440, 2010

BIOGRAPHIES

Ms. Ronica Raj is working as an Assistant Professor in Rizvi College of Engineering. She has completed Masters of Engineering in Computer Engineering from Thakur College of Engineering.

Professor Rukhsar Haji is Professor in Rizvi College of Engineering which is affiliated with Mumbai University. She has done Masters in Computer Engineering from Thadomal College. She have 6 years of teaching experience.

Ms.Farheen Rizvi pursued Diploma in Computer Engineering from Government Polytechnic Mumbai which is an Autonomous Institute. She is currently pursuing Bachelor of Engineering in Computer Engineering from Rizvi College of Engineering which is affiliated with the Mumbai University.

Ms. Naushin Khan is pursuing Bachelor of Engineering in Computer Engineering from Rizvi college of Engineering which is affiliated with the Mumbai University. She is aiming to pursue web page development for further studies.