



COMPARATIVE STUDY FOR SOFTWARE DEVELOPMENT IN COCOMO1 AND COCOMO2

¹ Dr. Rais Abdul Hamid Khan 1st Author, ² Garima Narendra Khatri 2nd Author, ³ Aarti Ganesh More 3rd Author, ⁴Utkarsha Pramod Mahajan 4th Author,

¹ Professor, SOCSE, Sandip University Nashik 1st Author, ^{2,3} BTech Scholar,

School of Computer Science and Engineering 2nd, 3rd Author, Sandip University, Nashik, INDIA

Abstract: Here's a abstract for a research paper on the comparative studies for software development in cocomo1 and cocomo2.

The comparison of the three best cost estimation models to be simulated for two prestigious organization datasets. It implies estimating the cost of all activities including software development, design, supervision, maintenance and so on. From online services to construction industry the field has spread and modified itself with the changes. Now the cost estimation industry has grown well. 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. 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.

Keywords: COCOMO model, COCOMO II, cost estimation, software engineering.

Introduction:

The software development industry has faced a continuous problem of the exact cost estimation with time. For overcome this problem a huge work and research has done till now, but still the estimation of the software is the core problem in any new project initiation. As I have gone through a few research

understandings, it is stated that there is still room for growing in the desired field. Numerous researchers have worked for the better results of the software industry in the developing of new projects. Many have developed their own models in the basis of a simple formulas with constant variables. Their constant variables are used for the variation in the development of a software for different backgrounds and purposes. Two types of research have been initiated i.e. application and system software. There is a fixed guideline for upgrading the industry and build a basics that contributes in the field, known as software engineering. It provides step by step solution and development of the software. Its method was to develop from physical to designing, testing, implementation and management of the system. Manalif et al presented model using expert COCOMO Model that provide information about project risk, estimated effort, and effort contingency required to identify risk.

COCOMO (Constructive Cost Model) is a well-known algorithmic software cost estimation model that has been used extensively in software development. COCOMO has undergone updates, with COCOMO I (1981) being the first version and COCOMO II (1995) being the updated version. Here's a comparative study between COCOMO I and COCOMO II. 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. 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.

From the studies and conclusion, all the models are best and were made for special purposes and all models are flexible and applicable to similar projects to be developed in future. It is very difficult for me to select those which have not compared yet. It is very difficult for me to select those which have not compared yet. In my work I had selected COCOMO-II, Halstead and IVR models to be compared for reducing the time and cost estimation of a software project to be developed. For this I am preferring the NASA 93 and TURKISH industry datasets.

From the result which shows that not only mixed software development approaches are used but also different effort estimation techniques are also tested, and those techniques mainly used the Fibonacci series as a measurement unit. Regarding the apparent benefits, all the participants agree with the listed group. The most important one is to drive the team to complete the project successfully. Similarly, the major perceived reason for the inaccuracy of the estimate is one of the required related issues called - Complexity and Uncertainty followed by Missing, changing and different requirements.

However, further enhance research should be conducted to gain a clear and broader more comprehensive understanding of this area. A successful New Product Development (NPD) must have three levers that must be in control: customer satisfaction, Time to Market (TTM) minimization and cost minimization. In all these complex circumstances where the market is competitive, the most crucial that contributes most is the cost. When designing a new venture of software development cost estimation is the critical factor in finding that whether the project is viable or not. Now the companies are most advance about the cost and confidence about the estimate in order to start a new project. Good estimate plays an important role in the performance and effect of the business. While underestimation plays financial loss to the business.

1. COCOMO1:

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.

COCOMO I is the original model, focusing on estimating the cost (in terms of effort) for software projects. It is based on a set of empirical data from completed software projects. COCOMO I consist of three levels of estimation: Basic, Intermediate, and Detailed. The Basic model provides an initial approximation based on size, while Intermediate and Detailed models allow for more precise estimations by adding additional cost drivers.

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 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 were proposed to help in providing a top-quality estimate to assist project manager in establishing accurate decision about their projects.

1) Basic Model:

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

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

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, 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:

- Planning and requirements
- System design
- Detailed design
- Module code and test
- Integration and test
- Cost Constructive model

2. COCOMO 2:

Numerous researchers have worked for the better results of the software industry in the developing of new projects. Many have developed their own models in the basis of a simple formulas with constant variables. Their constant variables are used for the variation in the development of a software for different backgrounds and purposes.

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.

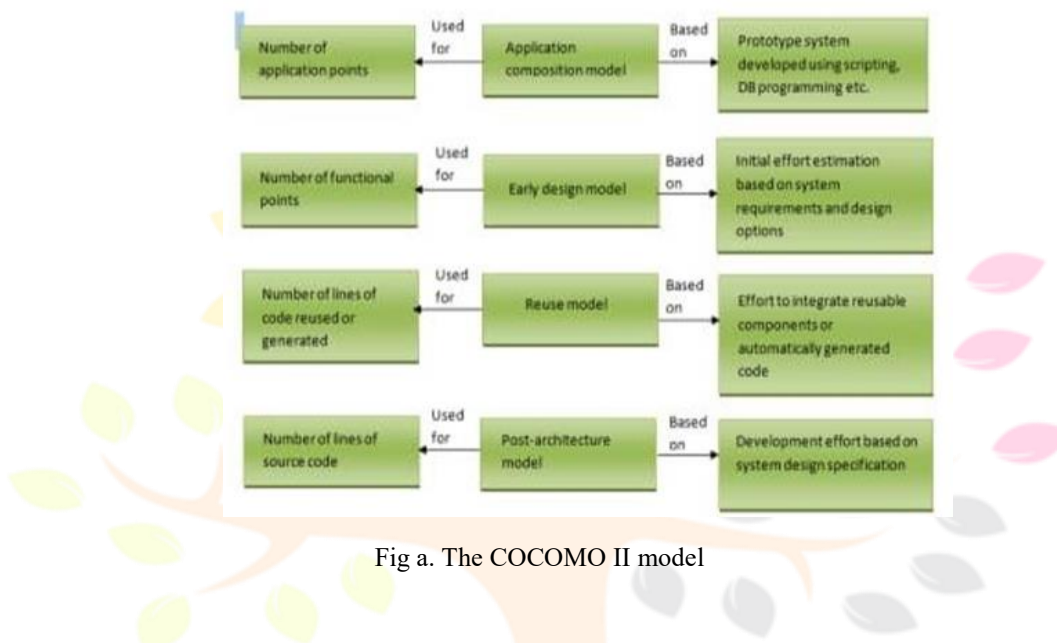


Fig a. The COCOMO II model

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.

This section briefly contains the simulation of all models in case studies. This can also leave the discussion in detail the preference of each model. Here it has thoroughly drawn an eye on the results comes out of the simulated model. This section can mainly cover the overall detail and the differences of all the models. Each model has their main attributes and terminology which has been briefly covered in this section. All the tables along with complete details and discussion are mentioned in this section of the research. The details of the dataset have been written in the form of case studies. Each model has been simulated separately and their results shown in the form of tables. From the simulation I can clearly in view that software cost estimation needs different attention. Every project has their own requirements. Every model is good in some circumstances but can vary changes for different requirement of the project. Overall COCOMMO II presents good results in comparing with Halstead and IVR.

COCOMO II estimates utilizes definitions of labour categories, thus they include project managers and program librarians, but exclude computer centre 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.

COMPARISION:

- COCOMO I is still useful for simple or traditional projects, particularly when a quick estimate is needed and there is little variability in project type.
- COCOMO II, with its enhancements, is the preferred model for modern software development. It offers improved accuracy, greater flexibility, and is more applicable to a wider range of project types, including those that follow iterative development, use modern tools, or have complex architectures.
- 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.
- COCOMO I effort equation's exponent is determined by 3 development modes.
- COCOMO II effort equation's exponent is determined by 5 scale factors.
- COCOMO I model is based upon the linear reuse formula.
- COCOMO II model is based upon the non-linear reuse formula.
- COCOMO I development begins with the requirements assigned to the software.
- COCOMO II follows a spiral type of development.

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.

Therefore, it was concluded from this study that based on standard Datasets Turkish Industry and NASA-93 COCOMO II is most significant than Halstead and IVR. Similar Study using Putnam model was conducted by Mittal.H. and P. Bhatia [20] and similar result was obtained. But the difference is that I use standard dataset Turkish industry, NASA-93 beside this it incorporates Halstead & IVR. Whereas, the previous study was based on local call manager android project, and they use Putnam model.

According to in the development process of a software project accurate time prediction is consider as one of the tough jobs. Because of a several reasons such that cost attribute, complexity of software project, staff programming skill time required for development of a project. At early stage when estimation is most crucial there is not enough information to provide accurate answer.

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 trade-offs 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 (COCOMO I, COCOMO 81).

REFERENCES:

“A DECISION SUPPORT SYSTEM FOR ESTIMATING COST OF SOFTWARE PROJECTS USING A HYBRID OF MULTI LAYER ARTIFICIAL NEURAL NETWORK AND DECISION TREE”, Javad Pashaei Barbin and Hassan Rashidi, International Journal in Foundations of Computer Science & Technology (IJFCST).

“A review paper on COCOMO model” Gajender pal Manish kumarand KuldeepbaralaDce, ggn.

<https://www.geeksforgeeks.org>.

Thamarai and S. Murugavalli, "Study of Modified Genetic Algorithm-Simulated Annealing for the Estimation of Software Effort and Cost", International Journal on Information Sciences and Computing.

A. Salam, A. Khan and S. Baseer, "A Comparative Study for Software Cost Estimation using COCOMO-II and Walston Felix Models", The 1st International Conference on Innovations in Computer Science & Software Engineering (ICONICS).

P.C. Pendharkar, "Probabilistic Estimation of Software Size and Effort", Expert Systems with Applications.

