A STUDY ON EFFORT ESTIMATION MODELS FOR THE SOFTWARE PROJECT MANAGEMENT

Akansha, Amit Kumar Chaudhary and Shrinivas Singh
Department of Computer Science & Engineering
Department of Computer Science & Engineering, Swami Vivekananda Subharti University, Meerut, U.P. India

ABSTRACT:

Accurately estimating the software size, cost, effort & development time or schedule is probably the biggest challenge facing software developers. Today. Estimating a project’s effort or schedule is a crucial task for software project management. Because, software project planning includes all of the activities i.e. estimation of how much money, how much time, how much effort, & how many resources, it will take to build a specific software based systems or product. Software Managers or Project Leaders are responsible for planning, scheduling, managing of the project development [1]. In this paper, a towards adaptive soft computing technique is explored to overcome the problems of uncertainty and inaccuracy resulting in improved process of software development effort estimation. So, fuzzy logic is applied to different parameters of Constructive Cost Model (COCOMO) II. The validation of results is carried out on COCOMO dataset. This paper’s contribution lies in its application of COCOMO MODEL mechanisms to the domain of software project management and presents accuracy for effort estimation methods. An initial evaluation among software professionals showed promising results and disclosed helpful hints for further development.

Keywords: Software Project Estimation, Software Cost Estimation, COCOMO, Fuzzy Logic.

[1] INTRODUCTION

After, the finalization of SRS, we would like to estimate the size, cost, effort & development time or schedule of the Project. By the software engineering, in the project management, main goal is to deliver the products on time with the desired quality attributes at the specified cost and the estimated effort. Estimation of these things are very necessary for attempt to software project management. Software development effort estimation is among one of the most challenging task that software developers need to perform, accurately. However, increasing competition among software companies, some managers give the fixed price projects but result in project failure. So, it’s very important to estimate the effort required size, schedule of a specified project ever in the early stages of software development. Project planning and control needs of management, Help set staffing needs and set budget and
schedule. Despite considerable research, it is still challenge to understand and predict what happens in a large software projects. Even in cases where some parties have good understanding of consequences, the business pressures and lack of quantitative evidence often results in misguided effort and faulty plans. An accurate cost estimation leads to effective control of time and budget during software development. To have an accurate estimate various models have been proposed in the past. Among those Boehm’s COCOMO is the most commonly used because of its simplicity for estimating the effort in person-month for a project at different stages of development.

[1.1] SOFTWARE EFFORT ESTIMATION MODELS

Cost estimation is one of the most challenging tasks in project management. Its purpose is to accurately estimate the resources needed and required schedules for software development projects. Software effort estimation models are divided into two main categories: algorithmic models and non-algorithmic models.

i. Algorithmic Method

The Function Point Method, COCOMO Model, T-shirt sizing, various estimation techniques

ii. Non-Algorithmic Method

New Approaches such as, Parkinson, Expert Judgment, Price-to-Win and Machine Learning Approaches. The principle of soft computing is to device methods of computation that leads to acceptable low cost solution, to an imprecisely/precisely stated problem. The soft computing techniques include methodologies like artificial neural networks, fuzzy logic, Bayesian networks and evolutionary computing. Due to their inherent nature these techniques are used to handle imprecision and uncertainty [14]. These techniques handle real life vague situations by providing flexible information processing capability [16]. Although, there is large amount of literature on software estimation methods(). There is often no detailed knowledge about when to apply which method. An estimation model for computer software use Empirically derived formulas to predict effort as a function of LOC/FP. Fuzzy Logic with its offerings of a powerful linguistic representation can represent imprecision in inputs and outputs, while providing a more expert knowledge based approach to model building. In this paper a fuzzy logic based COCOMO II model is proposed to so as to overcome the problem of imprecision and uncertainty.

COCOMO MODEL DESCRIPTION

COCOMO stands for Constructive Cost Model. The word "constructive" implies that the complexity of the model can be understood because of the openness of the model, which permits exactly to know WHY the model gives the estimates it does. The very first COCOMO Model was published by Dr. Barry Boehm in 1981, and reflected the software development practices of these days. COCOMO II provides a family (COCOMO suite) of increasingly detailed software cost estimation models, COCOMO II addresses the following three phases of the spiral life cycle: [5][6] applications development, early design and post
The COCOMO II model can be used to estimate effort and schedule for the whole project or for a project that consists of multiple modules. This paper contributes to software project manager to follow the particular effort estimation model. COCOMO II is actually a hierarchy of estimation models that address the following areas:

1) **Application composition model:** Used during the early stages of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount.

2) **Early design stage model:** Used once requirements have been stabilized and basic software architecture has been established.

3) **Post-architecture-stage model:** Used during the construction of the software.

**FUZZY LOGIC**

The Fuzzy Logic tool was introduced in 1965, by Lotfi Zadeh, and is a mathematical tool for dealing with uncertainty. It provides a technique to deal with imprecision and information granularity. The fuzzy theory provides a mechanism for representing linguistic constructs such as “many”, “low”, “medium,” “often,” “few”[20]. In general, the fuzzy logic provides an inference structure that enables appropriate human reasoning capabilities. On the contrary, the traditional binary set theory describes crisp events, events that either do or do not occur. It uses probability theory to explain if an event will occur, measuring the chance with which a given event is expected to occur [26].

![Fuzzy Logic Diagram](image)

**[2] RELATED WORK**

For, accurate estimation methods like i.e. For conducting/managing a successful software project should follow some basic things are following:-

1. Scope of work to be done.

2. The risk to be increased.

3. The resource are required.

4. The task to be accomplished.

5. The cost to be expanded.
6. The schedule to be followed.

There are two basic models for estimating software development effort (or cost)[16]:-

**Holistic** and **Activity-based**. The single biggest cost driver in either model is the estimated project size. Holistic models are useful for organizations that are new to software development, or that do not have baseline data available from previous projects to determine labour rates for the various development activities. Estimates produced with activity-based models are more likely to be accurate, as they are based on the software development rates common to each organization. Holistic models relate size, effort, and (sometimes) schedule by applying equations to determine the overall cost, and then applying a percent of the overall cost to each development activity. They do not consider the actual labour rates and costs of each activity.

**Popular holistic models include the following:**

1. SDM (Software Development Model - Putnam - 1978)
2. SLIM (Software Lifecycle Management - Putnam - 1979)
3. COCOMO (Constructive Cost Model - Boehm - 1981)
4. COPMO (Cooperative Programming Model - Conte, Dunsmuir, Shen- 1986)

In which above these models, COCOMO is most widely used Model.

**BASIC COCOMO**: COCOMO comes in three levels (basic, intermediate, and detailed) with each providing progressively more accurate estimates. This section gives a brief overview of basic COCOMO. Basic COCOMO is provided for three operational modes: organic, semi-detached, and embedded[13]. You would apply the organic mode to projects that have a small, experienced development team which is developing familiar applications in a familiar environment. You apply the embedded mode to large projects, especially when the project is unfamiliar or there are severe time constraints. The semi-detached mode is for projects somewhere in between. COCOMO predicts the effort and schedule for a software product development based on inputs relating to the size of the software and a no. of cost drivers that affect the productivity. “COCOMO MODEL [2] based on the 3-different modes that reflects the complexity.”

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

COCOMOII is actually a hierarchy of estimation models that address the following areas:-

1. Application Composition Model
2. Early design stage model
3. Post-architecture stage model

Estimate the effort of the project by this equation:-

\[
\text{Effort}(E) = C(SIZE)^K
\]

\[
E = a*(KLOC)^b
\]

\[
D = c*(E)^d
\]
Where, \( a, b, c, d \) -are the constants, \( E \)-effort, \( D \)-development time, \( P \)-productivity

\[
P=E/D
\]

<table>
<thead>
<tr>
<th>BASIC COCOMO</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORGANIC</td>
<td>2.4</td>
<td>1.05</td>
<td>2.5</td>
<td>0.38</td>
</tr>
<tr>
<td>SEMI-DETACHED</td>
<td>3.0</td>
<td>1.12</td>
<td>2.5</td>
<td>0.35</td>
</tr>
<tr>
<td>EMBEDDED</td>
<td>3.6</td>
<td>1.20</td>
<td>2.5</td>
<td>0.32</td>
</tr>
</tbody>
</table>

By the Boehm's Simple Model

\[
E=3.2*(KLOC)^{1.05}
\]

ADVANTAGES

- COCOMO is transparent; one can see how it works unlike other models such as SLIM.
- Drivers are particularly helpful to the estimator to understand the impact of different factors that affect project costs.

DRAWBACKS

- It is hard to accurately estimate KDSI early on in the project, when most effort estimates are required.
- KDSI, actually, is not a size measure it is a length measure.
- Success depends largely on tuning the model to the needs of the organization, using historical data which is not always available.

[3] PROBLEM STATEMENT

- Algorithmic COCOMO, COCOMOII failed to present or get the suitable solutions, not shows accuracy, independency, and certainty in every condition. Because, they are mostly used for particular type of project, environment, and does show the flexibility, Higher Reliability for the new system, new Platform etc.
- It is unable to deal with exceptional conditions, such as exceptional personnel in any software cost estimating exercises, exceptional teamwork, and an exceptional match between skill-levels and tasks.
- Poor sizing inputs and inaccurate cost driver rating will result in inaccurate estimation.
- Some experience and factors cannot be easily quantified.

Now a days there are many techniques are used for software estimation, here are we used fuzzy techniques to solve the of estimation such as cost, effort etc. Becoz we already know that estimation process is a very complex concept. And the main thing is for software project management.

The main objective of our thesis is to give estimated effort acc. To client needs, and used fuzzy Gaussian membership function, gives more accurate result.

[4] PROPOSED WORK

Akansha, Amit Kumar Chaudhary and Shrinivas Singh
We already know that COCOMO MODEL is used for estimating the effort and cost also. But for the future reference we must to apply such technique to get better. Fuzzy logic-based cost estimation models are more appropriate Method. In this paper, it is proposed to extend the Constructive Cost Model (COCOMO) [3] by incorporating the concept of fuzziness into the measurements of cost drivers; fuzzy set theory is used rather than classical intervals to represent the linguistic values. The advantages of this over quantization are that they are more natural and they mimic the way in which humans interpret linguistic values. It is important to stress that uncertainty at the input level of COCOMO model results in uncertainty at output [15]. COCOMO II comprises of size, cost drivers and scale factors input and effort as output which is measured in person months (PM)[14].The problem with software effort estimation is that it largely depends upon single values of size, cost drivers and scale factors. The size of the project is estimated based upon previously completed projects that are somewhat similar with the current project. Also cost drivers and scale factors need to have through assessment rather than assigning a fixed numeric value. To overcome this situation it would be better to represent these inputs in the form of fuzzy sets, in which interval values are, used which are represented by membership function. In the proposed model COCOMO II’s input parameters i.e. size, cost drivers and scale factors are taken into consideration to overcome the problem of effort estimation use fuzzy logic Gaussian membership function.

**Fuzzy sets:** A fuzzy set is a set with a smooth boundary.

**FUZZY MEMBERSHIP FUNCTION**

A fuzzy set is defined by a function that maps objects in a domain of concern into their membership value in a set. Such a function is called the membership function. Various Types of Membership Functions some are following- S-shaped function, Z-shaped function, Triangular Membership Function, Trapezoidal Membership Function, Gaussian Distribution Function, Generalized Bell Shaped Membership Function, Pi function Vicinity function.

**Gaussian membership function:**

\[ \mu(x, a, b) = e^{-\frac{(x-b)^2}{2a^2}} \]

The graph given in Fig. is for parameters \( a = 0.22, b = 0.78 \)
using GMF, the accuracy of effort estimation can be improved and the estimated effort can be very close to the actual effort.

**There are some Methods for Determining Membership Functions:**

Membership functions can be designed by analyzing the problem in hand. There are many possible forms of membership functions. Most of the actual fuzzy control operations are drawn from a small set of different curves. The methods for determining membership functions may be broadly classified into the following categories which are explained briefly as follows:

1. **Subjective evaluation and elicitation**
2. **Converted frequencies or probabilities**
3. **Physical measurement**
4. **Learning and adaptation**

The proposed fuzzy based software effort estimation model rules contain linguistic variables related to the project. The rule base for fuzzy inference system (FIS) make use of connectives ‘and/or’ for COCOMO input variables to form number of rules [22][23].

**[5] RESULTS**

The validation of the proposed model is carried out on a subset of projects from a repository of COCOMO dataset. A subset of dataset is applied to the proposed fuzzy model i.e. software development effort obtained using COCOMOII and efforts obtained by using fuzzy logic using membership function, are calculated. The conversion of dataset of COCOMO81 to COCOMOII is done [29]. It is observed that the effort obtained after applying fuzzy logic was closer to actual effort as compared to COCOMO II. There are many evaluation criteria for software effort estimation introduced in the literature, among them we applied the most frequent evaluation criteria such as: Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE). The parameter used for evaluation of proposed model is MRE and is given by:

$$MRE = \frac{100 \times |\text{actual effort} - \text{estimation effort}|}{\text{actual effort}}$$
MRE is calculated for COCOMO II as well as for the proposed fuzzy Gaussian Membership Function. It is observed that MRE obtained for the proposed model is quite less as compared to MRE obtained by COCOMO II. By the experimental research compare the effort which is based on cocomo model and fuzzy logic method. Take some values of the variables and find out effort by different method i.e. described above. The GMF that has been proposed in this work gives accurate effort than by using any other membership functions. it demonstrates a smoother transition between its intervals. The results clearly indicate that such fuzzy set modeling approach affects significantly the estimation outcomes.

[6] DISCUSSION & CONCLUSION

Three aspects of the problem have to be considered. First, is the estimation of size, which has to be accurate if any model of effort estimation will yield correct estimates of effort and schedule. Second, the manner of estimating effort from size (i.e. the formula used to predict effort necessary to complete the project). Third, the manner of estimating schedules (i.e. the formula for outlining a schedule when effort is known).

Based on the available data, we can conclude that the process of determining effort is quite accurate though it has not been possible to estimate schedules in the same way. So change requests are entertained from clients, due to lack of proper infrastructure, personnel availability and training, and mistakes of those who make initial estimates. Effort and schedule are very closely related to each other. While, the estimation of effort has been accurate the schedule estimates slip. This means that the effective person-week is not the one that has been assumed in the past. The actual number of hours of effective work done per day is fewer than expected due to the various specific external constraints operating in the context of this organization. The problem has to be tackled in two ways. Firstly, make all future estimates of the schedule based on the current effective and real person-week. This will help maintaining project schedules in the short run.

Software project is an important part of the software development process. The COCOMO suite (COCOMO II model and its extensions) offers a powerful instrument to predict software effort and cost also. Unfortunately not all of the extensions are already calibrated and therefore still experimental. It supports process improvement analyses, tool purchases, architecture changes, component make/buy tradeoffs and decision making process with credible results. Many endeavors were done to measure up to the changes in software life cycles, technologies, components, tools, notations and organizational cultures since the first version of COCOMO (COCOMOI, COCOMO 81)[25]. So, The study reveals that the proposed fuzzy logic based COCOMO II model overcomes the uncertainty in the inputs that is present in the traditional COCOMO and thus improves the accuracy of software effort estimation. Fuzzy logic provide the membership function to get accurate effort/cost besides its, it is not possible and also define flexibility in the output.by this we estimates the effort very flexibly and also reliability in the result.
REFERENCES


AUTHOR’S DESCRIPTION

Akansha
M.Tech Student of Computer Science & Engineering Department
Department of computer Science & Engineering Swami Vivekananda Subharti University, Meerut, U.P.
India

Amit Kumar Chaudhary
Assistant Professor of Computer Science & Engineering Department
Department of computer Science & Engineering Swami Vivekananda Subharti University, Meerut, U.P.
India

Shrinivas Singh
Senior Lecturer of Computer Science & Engineering Department
Department of computer Science & Engineering Swami Vivekananda Subharti University, Meerut, U.P.
India