IJARIIT
INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

A Review Paper on Software Cost Estimation using Fuzzy Logic

Nishi
nishisinghal54@gmail.com
Bhagat Phool Singh Mahila Vishwavidyalaya, Sonepat, Haryana

Vikas Malik
Vikassmalik@gmail.com
Bhagat Phool Singh Mahila Vishwavidyalaya, Sonepat, Haryana

ABSTRACT

With the pace of time software cost estimation is one of the most tedious works in project management. The entire process of estimation is uncertain in nature because it relies upon some attributes which cannot be predicted during the early stages of development. After studying various research papers given by different researcher we try to explore soft computing techniques which can be able to overcome the uncertainty and imprecision in estimation. The crucial focus of this article is to investigate the importance of fuzzy logic technique in enhancing the effort estimation accuracy, using COCOMO II by characterizing input parameters using various functions, for example, Gaussian, trapezoidal and triangular membership functions and comparing their results. In our study NASA (93) dataset will also be used in the evaluation of the proposed Fuzzy Logic COCOMO II. After that deep analysis will be done by various membership functions and then will declare which function gives better results for maximum criteria when compared with the other technique.

Keywords: *MRE, COCOMO, Fuzzy, Effort Multiplier, Software, Risk.*

1. INTRODUCTION

Software project failures have been an important subject in the last decade. Software Cost estimation predicts the cost of the resources which are required to execute all of the work of the software project. Uncertainties are referred as a risk.Risk is a measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule and performance constraints. Risk can be associated with all aspects of a program as these aspects relate to the Work Breakdown Structure and Integrated Master Schedule. Software cost estimation is a prediction of the cost of the resources that will be required to complete all of the work of the software project [1]. Large-scale software projects tend to have a high frequency of schedule overruns cost overruns, problems regarding quality, and outright cancellations. Instead of its bad reputation, it is important to note that some large software projects are finished on time, stay within their budgets, and operate successfully when deployed [2].

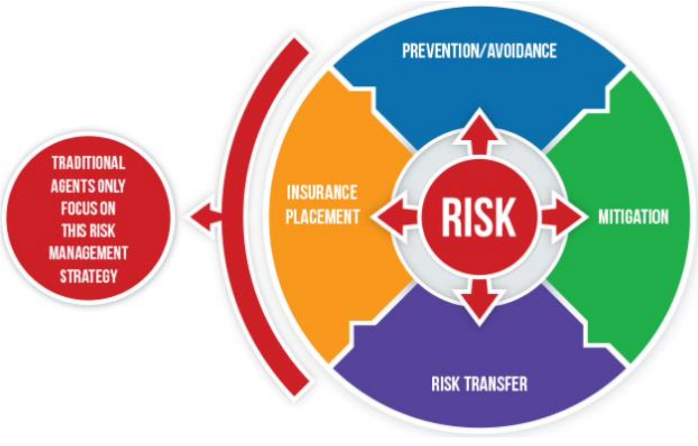

Figure 1 Risk Management Process

The importance of software cost estimation has been growing exponentially over last two to three decades. Software cost estimation is related to resources like time and manpower required to complete a software project. Software cost estimation starts at the proposed state and continues throughout the lifetime of a project. The estimation process includes various parameters for example effort estimation, size estimation, developing initial project schedules and finally estimating the overall cost of the project [3-4]. Therefore, in order to manage budget and schedule of software projects, various software cost estimation models have been developed. Accurate software cost estimates are critical to both developers and customers. The risk is a measure of future undetermined in achieving program performance aim and objectives within defined range like cost, schedule and performance constraints. Risk can be associated with all aspects of a program (e.g., threat, technology maturity, supplier capability, design maturation, performance against plan,) as these aspects relate across the Work Breakdown Structure (WBS) and Integrated Master Schedule (IMS).

2. Literature Survey

***(A): SEL Model*:** the University of Maryland which is known as SEL has established a model i.e. SEL Model for estimation. According to SEL model Estimation of effort, is defined as follows:

$$\text{EFFORT} = 1.4 * (\text{Size})^{0.93}, \text{ Duration D} = 4.6\ (\text{KLOC})^{0.26}$$

Effort (Person-Months) and lines of code (size in thousands of lines of code i.e. KLOC) are used as predictors.

(B): Walston-Felix Model: In 1977 Felix and Walston developed their own effort model from different aspects of the software development environment such as user database of sixty projects collected in IBM's Federal Systems division. It provides a relationship between delivering lines of source code. This model consists participation, changes which are customer-oriented, memory. According to Walston and Felix model effort is given by:

$$\text{EFFORT} = 5.2\ (\text{KLOC})\ 0.91, \text{ Duration D} = 4.1\ (\text{KLOC})0.36$$

In the year 2003, K.S Patnaik and S. Malhotra performed a work, “Software Development Effort Estimation using CBR: A Review”. In this research, the researcher found that the MMRE found to be nearly 32%, which is quite acceptable compared to the MMRE using regression analysis (approx. 90%). Further estimation by analogy helps in circumstances where it is not possible to generate an algorithmic model, such as Function Points [5]. In the year 2006, Elena Paslaru Bontas, Malgorzata Mochol performed a work “Ontology Engineering Cost Estimation with ONTOCOM”. This paper shows that Reliable methods for cost estimation are a fundamental requirement for a wide scale dissemination of ontologies in business contexts [6]. In the year 2012, Luiz Fernando Capretz performed a work, “Fuzzy-ExCOM Software Project Risk Assessment”. This paper shows that fuzzy ExCOM provides an efficient risk assessment with a higher sensitivity in risk identification, analysis, and prioritization as compared to the original Expert COCOMO methodology [7]. In the year 2013, Sweta Kumari and Shashank Pushkar performed a work, “Performance Analysis of the Software Cost Estimation Methods: A Review”. In this paper, a comprehensive overview of different types of software cost estimation methods and also describes the advantages and disadvantages of these methods [8]. This paper also presents some of the relevant reasons that cause inaccurate estimation. To produce a meaningful and reliable estimate, we must improve our understanding of software project attributes and their causal relationships, develop effective ways of measuring software complexity and the cost estimating process needs to be thoroughly arranged and carefully followed.

3. Planning of Work/Methodology

COCOMO provides an estimate of effort for a software project using a single property. The property used for the calculation of the corresponding estimate is the size of the software. This property is expressed in terms of KLOC (Kilo Lines of Code) and is calculated using the following formula:

$$\text{Effort} = X \times (\text{Size})^{y}$$

X, Y: Constants (depend upon the software project mode).

The intermediate COCOMO is sort of a wrapper level for the basic COCOMO and it uses the calculated values from the basic COCOMO adds some additional parameters to the estimation formula and recalculates the estimation [9]. According to the complete COCOMO, a software system is composed of various components or subsystems where each of these components or subsystems has different attributes from each other. COCOMO-II was published in 1995 having three sub-modes; an application-composition model, an early design model and a post-architecture model. COCOMO-II has, a set of seventeen Effort Multipliers (EM) or cost drivers as an input, which is used to adjust the nominal effort (PM) to reflect the software product being developed [10].

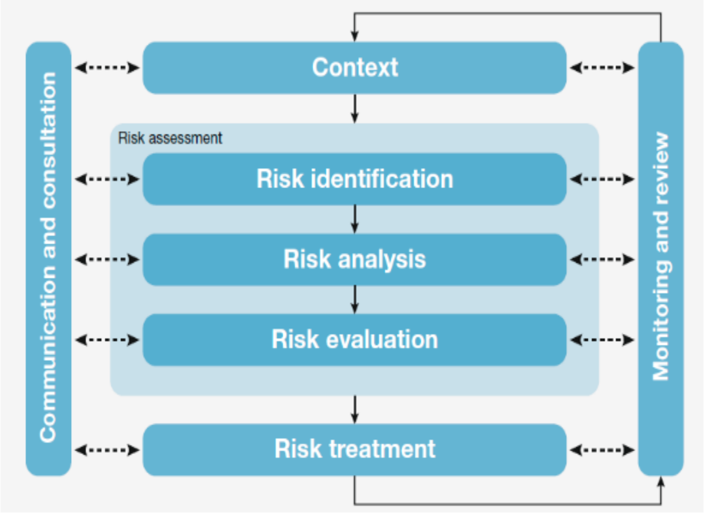

Figure 2 Risk Management Cycle

The concept of fuzzy logic (FL) is not a bearing methodology, however, could be a means of process knowledge by permitting partial set membership rather than a crisp set membership or non-membership. It supports the fuzzy set theory. Fuzzy systems are knowledge-based or rule-based system. The heart of fuzzy systems is a knowledge base consisting of the so-called Fuzzy "If-Then rules" in which some words are characterized by continuous member functions. The fuzzy logic systems can be divided into three types: pure fuzzy logic system, Takagi and Sugeno's fuzzy system and fuzzy logic system with fuzzifier and defuzzifier. Since most of the engineering applications produce crisp data as input and expects crisp data as output, the last type is the most widely used fuzzy logic systems with fuzzifier and defuzzifier. It was first proposed by Mamdani. It has been successfully applied to a variety of industrial processes and consumer products as shown in Fig3 below:

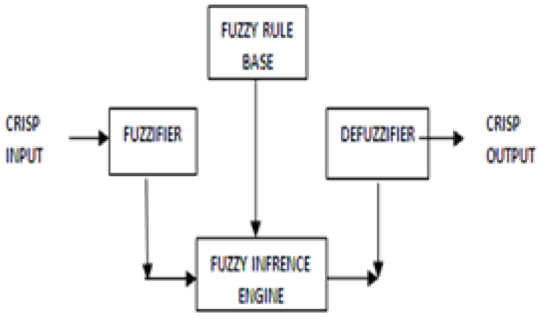

Fig.3 Fuzzy Logic System

Fuzzifier-It converts the crisp input into a fuzzy set, and to describe situation graphically, membership functions are used.

Fuzzy Rule Base-It uses "if-then rules" formulae.

Fuzzy Inference Engine- Bundle of if-then rules accumulated in fuzzy rule base is known as inference engine. It performs mainly two functions aggregation and composition.

Defuzzification- It converts the fuzzy output into the crisp output

4. Software used

Software: MATLAB Version R2015a: It is advanced software that provides an environment for numerical computation as well as graphical user interface display of outputs. In Matlab, the data input is in the ASCII format as well as binary format.

- Acquisition, Data Exploration, Analysing &Visualization
- Engineering complex drawing and scientific graphics
- Analysing of algorithmic designing
- Mathematical and Computational functions
- Modeling and simulating problems prototyping
- GUI (graphical user interface) building environment.

Using MATLAB, you can solve technical computing problems very easily and time-saving as compared to traditional programming languages, such as C, C++, and FORTRAN.

5. CONCLUSION

With the pace of time day by day advanced technology floored in the market. Today our main focus is to compete with advanced technology, that's why we try to develop such an advanced research which will be easy to use, more accurate, cost-effective and reliable. Some time back in the process of software development one issue is very crucial is an accurate and reliable estimation of the cost of software, manpower and time. Nowadays, in this research area, we use a fuzzy logic toolbox which is fourth-generation technology. Fuzzy logic can overcome the uncertainty and vagueness of software attributes. The aim of this paper was to find out the various application of applying fuzzy logic in software cost estimation that can be executed accurately so that final result would be optimized. In our research work, two parameters will be calculated which are a mean magnitude relative error (MMRE) and magnitude relative error (MRE) using various membership functions for different parameters. Also, the effort is calculated using various membership functions (MF) and compared the result based on the MMRE and PRED (25%) obtained for each of the membership functions.

6. REFERENCES

[1] Agarwal.R, Kumar.M, Malick.S, Bharadwaj.R.M, and Anantwar.D, “Estimating Software projects”, ACM SIGSOFT, Vol 26, 2001.

[2] Basavaraj. M..J, Dr. Shet.K.C, “Software Estimation using Function Point Analysis Difficulties and Research Challenges”, T.Sobh (ED.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, PP 111-116. © 2007 Springer.

[3] Azath. H, Wahidabanu.R.S.D, “Function Point: A Quality Loom for the Effort Assessment of Software Systems”, International Journal of Computer Science and Network Security, IJCSNS Vol.8 No12, Dec 2008

[4] A. J. Albrecht, and J. E. Gaffney, "Software function, source lines of codes, and development effort prediction: a software science validation", IEEE Trans Software Eng. SE-9, 1983,pp.639-648.

[5] K.S Patnaik and S. Malhotra performed a work, “Software Development Effort Estimation using CBR: A Review” March 2003

[6] Elena Paslaru Bontas, Malgorzata Mochol , “Ontology Engineering Cost Estimation with ONTOCOM” February 7, 2006

[7] Luiz Fernando Capretz performed a work, “Fuzzy-ExCOM Software Project Risk Assessment” 12 Dec 2012

[8] Sweta Kumari and Shashank Pushkar, “Performance Analysis of the Software Cost Estimation Methods, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013

[9] N. E. Fenton and S. L. Pfleeger, Software Metrics: A Rigorous and Practical Approach, PWS Publishing Company, 1997.

[10] G. R. Finnie, G. E. Wittig, AI tools for software development effort estimation, Software Engineering and Education and Practice Conference, IEEE Computer Society Press, pp. 346-353, 1996.

[11] D. R. Jeffery, G. C. Low, and M. Barnes, “A comparison of function point counting techniques”, IEEE Trans on Soft. Eng., vol. 19, no. 5, 1993, pp. 529-532.