A Study on Integrated Monitoring Software for Cost Estimation for Sustainability

Dr. S. SRIDHAR
Professor & DEAN
RV Centre for Cognitive & Central Computing Facility
R.V. College of Engineering, Mysore Road, Bangalore-560 059

Abstract: Software project management is the use of knowledge, proficiency, tools and techniques to project activities and gather project requirements. The success of Software Project Management (SPM) involves thorough planning, scheduling and budgeting. Project planning consists of actions performed to ascertain the total scope of the effort, describe and refine the target and develop the course of actions needed to achieve the targets. Cost estimation and budgeting are prime components of project planning process. As software applications are functions of size and connotation, the need for accuracy in software cost estimating has also increased. Accurate cost estimation is critical for both developers and customers and crucial for the success of the project, in most economic way. A sound awareness on the various estimation methods, resources used in estimating, past database sources and formats of estimating form all the basis for the sophisticated methods of estimating. Cost estimation plays a critical role in predicting the effort for calculating the feasibility of the project based on the input costs. It is essential that the technique used to estimate the cost should produce accurate results, which is pivotal for decision making. This paper presents a review on several methods of cost estimation.

Key words: Cost estimation, Project planning, Scheduling, Budgeting

I. INTRODUCTION

Several methods are available in literature in the field of estimation of software effort, which falls into one of the three broad categories as per Shepperd et al., [25], [20]: a) Expert judgment b) Algorithmic models and c) Machine learning. Out of the three methods, algorithmic models, based on statistic and regression analysis to estimate effort, constitute a major proportion. Though there are multifarious models in vogue, Boehm’s Constructive Cost Model (COCOMO) [4], Putnam’s Software Lifecycle Management (SLIM) [17] and the Function Point models by Albrecht and Gaffney [2] are the most applied models for software effort estimation. These models heavily depend on the local factors such as Lines of Code (LOC), Function Points etc. However, independent studies by Kemerer [14], have reported the occurrence of error in this approach. The unsolved practical problems linked with productivity in software cost estimation continue to be a major challenge for software professionals as per Lewis [15]. The problems in cost estimation are aggravated by the frequent changes in software technologies, which necessitate regular re-engineering of the models to stay on par with the current trends mentioned by Jones [11]. A successful and reliable software development project delivers a system which is accepted by the customers. The system can be termed successful if the owners of the software development and the users of the software are satisfied. It is invariably essential that the project manager needs to personally supervise the software development project for delivering the desired system as per the needs of the customer.

II. GOOD SOFTWARE COST MODEL

A good software cost estimation model assists the software project managers to take correct decision while managing the resources, planning and controlling the project and deliver the project as per time schedule and planned budget as per Chen and Menzies [6]. Martin Auer [19] proposed a model that deals with the study of analogy based cost estimation technique and provides sound estimates. However, this method does not handle the categorical datasets in an efficient manner. Sathyananda Reddy and Raju [23] claimed that fuzzy logic based cost estimation models are more useful when unclear and imprecision data is accounted. The advantage of this method is that they are more natural and analogous to the manner in which the human interprets the linguistic values. Though many membership functions are used in the available literature, Gaussian function outperforms other membership functions. The drawback of fuzzy method is that the imprecision and uncertainty are not dealt resourcefully. Ahmeda and Muzaffar [1] dealt with the imprecision and uncertainty in the inputs of effort prediction. Their research work presents a transparent, enhanced fuzzy logic based framework for software development effort prediction. Kazemifard et al [12] has proposed a novel method called FESCE where the team characteristics are involved in a team member agent system along with the project characteristics for finding the effort estimation using COCOMO II. The disadvantage of
this method is that it has not handled the neuroticism behaviour of a team member which is one of the important dimensions of personality trait. As per Silvia et al., [25] Personality refers to the characteristics of a person in an organization which differs from person to person. Personality is the component of psychology trait which is categorized into 5 dimensions, namely, 1) Neuroticism, 2) Extraversion, 3) Openness, 4) Agreeableness, and 5) Conscientiousness. Neuroticism is one of the significant components of personality trait which refers to the degree of a person in anxious, irritable, and moody state. Basically, the anxiety of a person and depression are directly proportional in a work environment, thereby leading to emotional troubles resulting in depressive state mentioned by Keith et al. [13].

III. SOFTWARE PROJECT

A software project is like a social system where personal and neuroticism characteristics play a stellar role to accomplish the targets. The project associates have emotions and neuroticism characters like anger, anxiety, joy, and skill, which can have positive or negative impact on the outcome of the project. The positive impact is related to joy and skill while anger and anxiety leads to negative trend. The present study utilizes neuroticism characteristics, anger, and anxiety, to assess their impact on the outcome of the project. Oren and Ghasem-Aghae formulated a standard model that personality can be represented in fuzzy logic for simulation of human behaviour claimed by Oren and Ghasem [21]. There are no definite and reliable models to estimate the effort in terms of the team characteristics like social and neuroticism characters though there are number of methods to ascertain the project features. An approach is made in this research to delineate the performance of neuroticism character on effort estimation. An approach, namely, FUZzy ANalogy Neuroticism (FUZANN) has been evolved, which encompasses the Fuzzy logic based analogy method with neuroticism characters to unveil the effort for cost estimation. This method is tested with the static and dynamic dataset (STDA) using historical and real time project data and the results indicate that it performs in a good manner while estimating the software effort and signifies the role of neuroticism character in producing accurate effort estimates.

IV. NATURE AND COMPLEXITY

The nature and complexity of software has changed considerably during the last thirty years. Development of Software is expensive and it is a major cost factor in any information system budget. The accuracy in the estimation of software project cost has a direct and major impact on the quality of the firm’s software investment decisions. L. Angelis et.al. [3] developed a multi-organizational software cost estimation model for analyzing the software cost data collected by the ISBSG. This model performs better than the conventional least squares regression models developed from the same database. Categorical regression, organization of data and issues in collecting software data are not clearly verified in the analysis process of this method. Magne Jorgensen and Kjetil Molokken-Ostvold [18] proposed an approach to find out the reason behind the occurrence of the software effort estimation error. They focused on the cause, rather than the estimation error and also discussed the functioning of previous methodologies. The authors concluded that estimation error is dependent on the role of the respondents, data collection and based on the analysis of the estimated error. Carolyn Mair and Martin Shepperd [5] presented a paper on the empirical comparisons between analogy and regression methods of software project cost estimation. The aim of this paper is to determine whether the results of the empirical studies of the software engineering cost estimation is consistent or not, based on the regression and other analogy techniques. The technique used to ensure the consistency of the results is exhaustive literature search, based on the predefined inclusion and exclusion criteria. However, the authors have concluded that only 25% of the studies were found to be internally conclusive. These methods were noticed to be less consistent in the final results because of inconsistent patterns comparing regression and analogy.

V. ENHANCED CASE-BASED REASONING

Hasan Al-Sakran [9] has put forward an enhanced Case-Based Reasoning (CBR) approach incorporated with multi-agent technology to recover similar projects from multi-organizational disseminated datasets. The study explores the opportunity of developing a software cost estimation model by collecting software cost data from disseminated and predefined project cost databases. The model applying CBR method can be used to find similar projects with historical data, derived from measured software projects developed by different organizations. Jo E. Hannay et.al. [8] have presented a paper that describes the systematic review of theory used in software engineering experiments. They suggest empirical software engineering for evaluating the type of development technology to find out their effects on the software development. Jacky Wai Keung et al., [10] suggested an extension of analogy to address the searching techniques of similar projects based on a sound statistical basis using the Euclidean distance measure for finding the nearest project selection, thereby removing the need for heuristic search and improving its algorithmic performance. However, this is not the best solution, when the
categorical variables are huge in number. Y.F. Li et al. [16] proposed the project selection technique for ABE (PSABE) which reduces the whole project base into small subsets and consists of only the representative projects. Moreover, PSABE has been combined with the feature weighting to form FWPSABE for a further improvement of ABE. The method was validated on four datasets (two real world sets and two artificial sets) and compared with conventional ABE, feature weighted ABE (FWABE) and machine learning methods. Wei Lin Du et al. [26] suggested an approach using an integrated neuro-fuzzy technique and the SEER-SEM effort estimation algorithm. Increment rating values and linguistic values are specified as inputs for evading the huge estimation deviation among similar projects. The performance of the model was accessed by designing and evaluating with published projects and industrial data. The results indicate that estimation with neuro-fuzzy model containing SEER-SEM was improved in comparison with the estimation results that only use SEERSEM algorithm. At the same time, the results also demonstrate that the general neuro-fuzzy framework can function with various algorithmic models for improving the performance of software effort estimation.

VI. ANALOGY-BASED EFFORT ESTIMATION

Ekrem Kocaguneli et al., [7] identified the essential assumption of analogy-based effort estimation: i.e. the immediate neighbours of a project offer stable conclusion about that project. They tested by creating a binary tree of groups of effort data and estimated the variance of super-trees versus smaller sub-trees. Roheet Bhatnagar and Mrinal Kanti Ghose [21] utilizes neural network and Mamdani FIS model to envisage the effort estimation by student dataset and concluded that Linear Regression Neural Network (LRNN) has the smallest value for MMRE in comparison with the other neural network models. However, when compared with fuzzy logic, it is pragmatic that fuzzy logic works in an admirable way with neural network models.

VII. CONCLUSIONS

This note has presented a good literature on the following:- The success of Software Project Management (SPM) involves thorough planning, scheduling and budgeting. Project planning consists of actions performed to ascertain the total scope of the effort, describe and refine the target and develop the course of actions needed to achieve the targets. Cost estimation and budgeting are prime components of project planning process. As software applications are functions of size and connotation, the need for accuracy in software cost estimating has also increased. Accurate cost estimation is critical for both developers and customers and crucial for the success of the project, in most economic way.

A sound awareness on the various estimation methods, resources used in estimating, past database sources and formats of estimating form all the basis for the sophisticated methods of estimating. Cost estimation plays a critical role in predicting the effort for calculating the feasibility of the project based on the input costs. It is essential that the technique used to estimate the cost should produce accurate results, which is pivotal for decision making. This paper will be very useful for researchers working in this domain.

VIII. REFERENCES


