Software Cost Estimation Research Decade: A Review

Karuna Shankar Awasthi[1], Anand Kumar Rai[2] and Laxmi Shankar Awasthi[3]

[1, 3]Deptt. of Computer Science, Lucknow Public College of Professional Studies, Lucknow.
[2]Deptt. of Computer Science, Mumtaz Post Graduate College, Lucknow.

ABSTRACT

Software Cost Estimation is a technique to predicting the cost of the software project which is proposed to develop. Cost estimation will remain a complex issue and researchers should be involved in implementing a new process for this work. The accuracy of the software project cost estimation has a direct and significant impact on the quality of the company's software investment decisions. Accurate cost estimation technique can reduce unnecessary costs and increase organizational efficiency. In this research, many types of estimation techniques have been proposed over the past 10 years. This paper focuses on short-term literature reviews of various cost-estimating techniques including the latest trends in the field.

KEYWORDS: Cost estimation, Cost drivers, Effort estimation, Expert judgment.

1. INTRODUCTION

It is estimated that about one-third of the projects exceeded its budget and arrived late and two-thirds of all major projects exceeded their original estimates. Accurate estimates of software development costs are critical to making good management decisions and accurately determining how much effort and time the project requires of project managers and program analysts and developers. After 20 years of research, there are a number of software cost estimates available that include algorithmic methods, analogy measurement, expert judgment method, winning method, top method, and bottom method. There is no better or worse way, in fact, their strengths and weaknesses often complement one another. Understanding their strengths and weaknesses is very important when you want to balance your projects [1].

1.1 Algorithmic methods

The algorithmic method is designed to provide specific mathematical calculations to perform software measurements. These statistics are based on research and historical and operational data such as Source Lines of Code (SLOC), the number of tasks to be performed, and other cost factors such as language, construction method, skill levels, risk assessment, etc. algorithmic methods have been widely studied and there are many advanced models, such as the COCOMO models, the Putnam models, and the operating point models [2].

1.1.1 Expert Judgment Method

The Expert judgment techniques include consulting with the expert group to exploit their knowledge and understanding of the project to attain estimate of the cost of the project. The strong point and weaknesses are balancing to the strengths and weaknesses of algorithmic method [2].

1.1.2 Estimating by Analogy

In this method the project cost estimation is done by comparing the proposed project to previously completed almost similar project. The real data from the completed projects are generalized to estimate the project [2].

1.1.3 Top-Down and Bottom-Up Methods

1.1.3.1 Top Down Method

The top-down model is also called the Macro Model. Using a top-to-bottom measurement method and the project is subdivided into various sub-standard components. The best way to use this method is the Putnam model. This method is most effective in estimating the initial cost when only international buildings are known. In the first stage of software development, it is very helpful because no detailed information is available.

1.1.3.2 Bottom-up Estimating Method

Using the bottom-up measurement method, the cost of each software component is estimated and aggregates results to evaluate the cost of the entire project. It aims to create system limitations from the accumulated information about minor software components and their interactions. The best way to use this method is a detailed model of COCOMO.

1.1.4 COCOMO Mode

The Construction Cost Model (COCOMO) a cost-effective software model widely used by the algorithmic software cost model. Estimates based on the COCOMO basic model can be made more accurate by considering other factors related to the required software development features, qualifications and experience of the development team, and the software development environment.

1.1.5 Putnam model

Putnam model is another very effective model for the software cost estimation. In this model, Technical constant C= size * B1/3 * T4/3, Total Person Months B=1/T4 *(size/C)3, where T= Required Development Time in years and Size is estimated in LOC. The Putnam model is very sensitive in the constant of development time.

1.1.6 Function Point Analysis Based Methods

This cost estimation analysis is another method of quantifying the size and complexity of the proposed software system in terms of the functions that the proposed project delivers to the user.

1.2.1 Neural Networks

Neural networks make available the prediction like human brain. With the help of certain training pattern of the neurons we can make prediction of any kind of complex system. A neuron encompasses a weighted amount of input and produces an output if the amount exceeds a certain limit. This output then becomes a positive (positive) or negative (negative) input to other neurons in the network. The process continues until one or more results are released. The Neural Network is known for its ability to deal with the problem of segregation [4].

1.2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a computer program that solves a problem by trying to improve the contestants' solution in terms of a certain level of quality. Such methods are popularly known as Meta Heuristics as they make little or no thought about the problem at hand and can search for very large solution spaces.

1.2.3 Genetic Programming

Genetic engineering is one of the evolutionary mechanisms for balancing effort. Evolutionary calculations are characterized by the fact that the solution lies in a generation of elite solutions determined by the conditions of 'survival of the fittest'. When GA is used to solve real-world problems, a population that contains a set of random people is created. The population was tested during the evolutionary process. Each person is given a measure, which indicates the degree of human adaptability to the environment. Percentages of highly modified people are retained, while others are discarded. People who are kept in the selection process can get a change in their basic features with a recycling machine. This method is used by current people who aim to explore the area of search and find better solutions to this problem through crossovers and transformers that make young people the next generation. This process, called reproduction, is repeated until a satisfactory solution is found [4].

1.2.4 Fuzzy Logic

Fuzzy logic is an important tool, which can be used to solve complex problems where a mathematical model is too complex or impossible to build. It has also been used to reduce the complexity of existing solutions and to increase the accessibility of control theory. Software development is always characterized by parameters with a certain degree of instability. Studies have shown that a complex logic model plays a role in the evaluation of software efforts. The use of abstract concepts is able to overcome some of the problems associated with existing measurement strategies. Fuzzy comprehension not only helps predict effort, but is essential for improving the quality of current measurement models. Fuzzy logic empowers language representation in the input and output of a model to tolerate processing. It is especially suitable for effort measurement as most of the software properties are measured in the standard or ordinal type which is a specific language value [9].

2. REVIEW OF MAJOR RESEARCH FROM 2002 TO 2012

In 2002, Jorgensen, suggested that expert estimation was the most commonly used successful cost estimation strategy for software projects. In 2003, Yunsik Ahn, Jungseok Suh, Seungryeol

Kim and Hyunsoo Kim, suggested SMPEEM (Software Maintenance Project Effort Estimation) focusing on the software maintenance effort and cost estimation [2][3].

In 2006, Hasan Al-Sakran, introduced hybrid software cost estimation model, which assimilates multi-agent technology and case-based reasoning. This study reveals the use of cased-based reasoning for the software cost estimation using multi organization databases combined with the mobile agent technology. this suggested method used for those projects which have no any historical data for the cost prediction.

Stein Grimstad, et. al, (2006) has taken a real example where the general analysis of the cost estimation error leads to an erroneous conclusion. It has been suggested that there is a need for a better analysis of software cost error. It was concluded that a checklist to identify non studied items that may have a potential bias for measuring error measurement, emphasis on appropriate measurement term, and support for isolation strategies is helpful [8].

In 2007, different effort estimation was introduced. In this period, it was proved that the average accuracy of expert judgment was higher than all existing cost estimation models.

In 2008, Parvinder S. Sandhu studied on the accuracy of different cost estimation models. In this study a soft computing approach, neuro-fuzzy system was advocated to create the model because Neuro-Fuzzy system was very effective able to predict the approximate the non-linear function with more accuracy.

Vahid Khatibi, Dayang N. A. Jawawi, (2011), Here most of the cost estimation strategies are systematically demonstrated. It has been found that there is no standardized method that can be used for best software cost estimation in all different situations and each procedure may be suitable for a specific project. It is necessary to understand the principals of each cost estimation method the best choice. Some test metrics and a real model of measurement are presented in this paper to explain the effectiveness of the estimationmethod [10].

In 2011, Zulkefli Mansor, Saadiah Yahya, Noor Habibah Hj Arshad, suggested and discussed the key factors that contribute to the traditional and agile process of measuring the cost of a software development project. In addition, this paper has introduced the key elements that lead to the success of traditional and cost-effective measurements in a software development project. In view of these factors, it has been found that agile cost estimation process can produce more accurate results and can reduce effort, time, and cost [11].

In 2012, Narendra Sharma, Ratnesh Litoriya introduced the cost drivers with the help of a data mining tool. These results suggest that the development of data mining techniques and machine learning strategies in existing software testing software such as COCOMO can effectively improve the performance of a proven method. While an excellent combination of data mining techniques were consistent with different data classification, it has been shown that there are different types of software projects and that complex data collection should continue to improve the development of accurate cost estimation models [7].

3. CONCLUSION AND FUTURE TREND

In this review paper, all the studies have been presented of a decade (2002-2012) and it has been analyzed that gradually different studies in the area of software cost estimation enhanced the accuracy of the software cost prediction. Use of AI and data science in the field of software engineering has been proved a very successful tool. Recently, increasing complexity of the software and hardware technology needs identification of more cost drivers. Therefore, more research are needed using the AI approach in the field of software cost estimation.

REFERENCES

[1] G. Wittig and G. Finnie, (1997). "Estimating Software Development Effort with Connectionist Models" Information and Software Technology, vol. 39, 1997, pp. 469-476.

[2] Hasan Al-Sakran, (2006), "Software Cost Estimation Model Based on Integration of Multi-agent and Case-Based Reasoning", Journal of Computer Science 2 (3), pp 276-282.

[3] H.M. Gunaydin and S. Z. Dogan,(2004), "A neural network approach for early cost estimation of structural systems of buildings," International Journal of Project Management, vol. 22, no. 7, pp 595–602,

[4] K. Srinivasan, and D. Fisher, (1995), "Machine Learning Approaches to Estimating Software Development Effort." IEEE Transactions on Software Engineering, vol. 21, no. 2. February, 1995, pp. 126-136.

[5] Martin Shepperd and Chris Schofield, (1997), "Estimating Software Project Effort Using Analogies", IEEE transactions on software engineering, vol. 23, no. 12, pp. 736-743.

[6] M. Jorgensen, M. Shepperd,(2007), "A Systematic Review of Software Development Cost Estimation Studies", IEEE Transactions on Software Engineering, Vol. 33, No. 1, IEEE Computer Press, Washington D.C., 2007, pp. 33-53.

[7] Narendra Sharma, Ratnesh Litoriya, (2012), "Incorporating Data Mining Techniques on Software Cost Estimation: Validation and Improvement", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 3,pp. 301-309.

[8] Stein Grimstad, et. al, (2006), "A Framework for the Analysis of Software Cost Estimation Accuracy", ISESE'06, ACM 1-59593-218- 6/06/0009.

[9] S. Huang and N. Chiu, (2009), "Applying fuzzy neural network to estimate software development effort", Proceedings of Applied Intelligence Journal, vol. 30, no. 2, pp. 73-83.

[10] Vahid Khatibi, Dayang N. A. Jawawi, (2011), "Software Cost Estimation Methods: A Review", Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No. 1, pp 21-29.

[11] Zulkefli Mansor, Saadiah Yahya, Noor Habibah Hj Arshad, (2011), "Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project", International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3), pp.942-952.