Proposing a New Estimation Model Based On Putnam Model

Anuj Khuttan[1], Archana Singh[2]

[1, 2]Galgotias University, School of computer Science and Engineering, Greater Noida, Uttar Pradesh, India

Abstract: ***Estimating Effort for producing software at early stage of the software life cycle is a very crucial task for any organization. Many organization use different techniques to evaluate effort required for producing software, at the different levels of software life cycle model. In this paper, we are applying effort multipliers to the Putnam model and comparing the results with the previously developed models like Putnam.***

Keywords: Effort, Putnam Model, Effort multipliers, artificial neural network

1. Introduction

The ability to accurately and consistently estimate software development efforts, especially in the early stages of the development life cycle, is required by the project managers in planning and conducting software development. Several estimates are involved to effectively manage the software Effort. Estimation has become necessary for any community to develop useful models that estimate effort accurately. Software development efforts estimation is the process of predicting the most realistic use of effort required to develop or maintain software based on incomplete, uncertain and/or noisy input. Putnam model is an empirical software effort estimation model developed by Lawrence H. Putnam. In this model effort estimates are made by providing size and calculating the associated effort [1]. ANN-COCOMO based software estimation neural networks proposed by ImanAttarzadeh and Siew Hock Ow, in their proposed neural network model, the accuracy of Effort estimation can be improved and the estimated cost can be very close to the actual Effort [4]. The model we are proposing is comprised of previously developed Putnam model and effort multipliers. Use of artificial neural network provides more accurate results, which are very close to actual effort.

2. Putnam Model

2.1 Putnam effort estimation (also known as SLIM)

The software equation is a dynamic multivariable model that assumes a specific distribution of effort over the life of a software development project. The model has been derived from productivity data collected for over 4000 contemporary software projects. Based on these data, an estimation model of the form.

$$E = [LOC * B^{0.333}/P]^3 * (1/t^4)$$

Where,
E =effort in person-months or person-years
t=project duration in months or years
B =“special skills factor”
P = “productivity parameter” that reflects:

- Overall process maturity and management practices
- The extent to which good software engineering practices are used
- The level of programming languages used
- The state of the software environment
- The skills and experience of the software team
- The complexity of the application

Typical values might be P = 2,000 for development of real-time embedded software; P = 10,000 for telecommunication and systems software; P = 28,000 for business systems applications. The productivity parameter can be derived for local conditions using historical data collected from past development efforts.

It is important to note that the software equation has two independent parameters: (1) an estimate of size (in LOC) and (2) an indication of project duration in calendar months or years.

To simplify the estimation process and use a more common form for their estimation model, Putnam and Myers suggest a set of equations derived from the software equation [5].

Minimum development time is defined a
$t_{min} = 8.14 (LOC/P)^{0.43}$ in months for $t_{min} > 6$ months
$E = 180 Bt^3$ in person-months for $E \geq 20$ person-months

Note that t in Equation is represented in years.

One significant problem with the PUTNAM model is that it is based on knowing, or being able to estimate accurately, the size (in lines of code) of the software to be developed. There is often great uncertainty in the software size. It may result in the inaccuracy of effort estimation [1].

3. Proposed Putnam Model

In Putnam Model we are applying Effort multipliers to the Putnam model and checking simulations on ANN. The purpose of choosing ANN is to evaluate effort more accurately as much as close to the actual effort. ANN is used to evaluate effort in person month. We are applying size, scaling factor, technology constant, effort multipliers and time as an input to the ANN. In order to structure the network to accomplish the Putnam post-architecture model, a specific hidden layer and a sigmoid activation function with some pre-processing of data for input layer is considered. The proposed network is not a fully connected network but specified hidden layer nodes that take into account the contribution of EM and SF separately.

$t_{min} = 8.14 \, (LOC/P)^{0.43}$ (in months for $t_{min} > 6$ months)

$E = 180 \, Bt^{3*} \prod_{i=1}^{15} EM_I$ (in person-months for $E \geq 20$ person-months)

Where, EM = effort multipliers

4. Results and Discussion

1) Data

To compare all the model discussed above we have to take data form COCOMO NASA2/ Software cost estimation which have the data of different centers, 93 NASA projects between years 1971-1987 was collected by JairusHihn, JPL, NASA, Manager SQIP Measurement & Benchmarking Element [11].

2) Evaluation Method

For evaluating the different software effort estimation models, the most widely accepted evaluation criteria are the mean magnitude of relative error (MMRE) and probability of a project having a relative error of less than or equal to 0.25 The Magnitude of Relative Error (MRE) is defined as follows:

$$MRE_i = \frac{|Actual\ Effort - predicted\ Effort|}{Actual\ effort}$$

The MRE value is calculated for each observation whose effort is predicted. The aggregation of MRE over multiple observations (N) can be achieved through the Mean

$$MMRE = \frac{1}{N} \sum_i^N MRE_I$$ [4]

3) Calculations

Effort calculated with the given NASA data

Sr. No.	*Size*	*Actual Effort*	*Putnam (in PM)*	*ANN Putnam (in PM)*
1	25.9KLOC	117.6	42.44	46.44
2	100KLOC	360	337	379.95
3	190KLOC	420	772.88	315.3
4	302KLOC	2400	1405	1803
5	350KLOC	720	1699.7	822.65

4) MRE

Sr. No.	*Model*	*MMRE*
1	Putnam	0.06389
	ANN Putnam	0.0554
2	Putnam	0.840
	ANN Putnam	0.025
3	Putnam	0.4146
	ANN Putnam	0.2487

5) MMRE

Sr. No.	*Model*	*MMRE*
1	Putnam	0.4395
	ANN Putnam	0.1097 %Improvement 24.96%

5. Conclusion

It is clear with our research is that Putnam model doesn't work for the small software size project. In our proposed model Putnam model is giving more accurate for medium size project as compared to the previously developed model. We can see % improvement over previously developed model is 24.96%.

References

[1] Putnam, L. H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem", IEEE Transactions on Software Engineering, Vol. 4, No. 4, pp. 345 – 361, 1978.

[2] Boehm B. W. "Software Engineering Economics", Englewood Cliffs, NJ, Prentice-Hall, 1981.

[3] Boehm B., Abts, C., and Chulani, S., Software Development Cost Estimation Approaches – A Survey, University of Southern California Center for Software Engineering, Technical Reports, USC-CSE- 2000-505, 2000.

[4] Imran Attarzadeh, Siew Hock Ow, "Purposing a New Software Cost Estimation Model Based on Artifical Neural Networks", IEEE, 2010.

[5] Roger S. Pressman, "Software Engineering: A Partitioner's Approach", MaGraw Hill, edition 5th, 2001.

[6] Farhad SOLEIMANIAN GHAREHCHOPOGH, "Neural Network Application in Software Cost Estimation: A Case Study", IEEE, 2011.

[7] B. Boehm, E. Horowitz, R. Madachy, D. Reifer, B. k. Clark, B. Steece, A. W. Brown, S. Chulani and C. Abts, "Software Cost Estimation with COCOMO II", Prentice Hall, 2000.

[8] G. Witting and G. Finnie, "Using Artificial Neural Networks and Function Points to Estimate 4GL Software Development Effort," Journal of Information Systems, vol. 1, no.2, 1994, pp. 87-94.

[9] N. Karunanitthi, D. Whitely, Y. K. Malaiya, "Using Neural Networks in Reliability Prediction," IEEE

Software Engineering, vol. 9, no.4, 2011, pp. 53-59.

[10] B. Boehm, C. Abts, and S. Chulani, "Software Development Cost Estimation Approaches – A Survey," University of Southern California Center for Software Engineering, Technical Reports, USC-CSE-2000- 505, 2000.

[11] Tim Menzies, "COCOMO NASA 2/Software Cost Estimation", Promise Software Engineering Repository (http://promise.site.uottawa.ca/SERepository/datasets-page.html), 3 April 2006.

[12] Putnam, L. H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem", IEEE Transactions on Software Engineering, Vol. 4, No. 4, pp. 345 – 361, 1978.

Author Profile

Anuj Khuttan, He is pursuing M.Tech, in Galgotias University Greater Noida, Delhi NCR, He is completed B.tech in Electronics And Communication from Guru Gobind Singh Indraprastha University, New Delhi, Area of interest is Software Engineering and Data mining

Archana Singh, She is pursuing M.Tech, in Galgotias University Greater Noida, Delhi NCR, She is completed M.Sc in Information Technology from Punjab Technical University, Jalundhar, Area of interest is Software engineering and Data Mining.