Elimination of Estimation biases in the Software Development

Thamarai . I.

Research scholar, computer science department
Sathyabama University, Chennai, India
ilango.thamarai@gmail.com

Dr.S.Murugavalli (*Co-Author*)

Supervisor, Computer Science Department
Sathyabama University, Chennai, India
Murugavalli26@ rediffmail.com

Abstract

The software effort estimations are usually too low and the prediction is also a very difficult task as software is intangible in nature. Also the estimation is based on the parameters that are usually partial in nature. It is an important management activity. Despite much research in this area, the accuracy of effort estimation is very low. This results in poor project planning and failure of many software projects. One of the reasons for this poor estimation is that the estimation given by the software developers are affected by some information which do not have any relevance to the calculation of effort. To avoid this, we have proposed a new methodology in which we analyze the relationship between the estimation bias and the various features of developers such as the role in the company, thinking style, experience, education, software development skills, etc. It is found that the estimation bias increases with higher levels of interdependence.

Keywords: Software Effort Estimation, Bias in estimation, Interdependence, software development.

1. Introduction

Software project planning is one of the most important activities in a software development process. Planning largely depends on the effort estimation and this requires many parameters such as size, number of persons, schedule, etc. It is a difficult and complicated task. Linda M. Laird enumerates the reasons for the inaccurate effort estimation [1]. Ning Nan and Donald E. Harter emphasize the role of budget and schedule pressure [2]. In [3], M. Jorgenson and D.I.K. Sjoberg demonstrated how the software effort is affected by the client's expectations about cost. Tim Menzies *et al* suggested to create clusters and proved that the best are that which are near the source data but not from the same source as the test data [4]. The general principles of software effort estimation were explored by Ekrem Kocaguneli *et al* to guide the design for estimation [5].

In [6], Magne Jorgenson and Stein Grimstad made a detailed study on how irrelevant information can affect the estimation of software. According to them, the field settings that led to the irrelevant information have a very small impact than the artificial settings for doing experiments. Studies have demonstrated the biases in human judgment [7]. When the biases are towards lower effort estimation, it may lead to problems in project planning. This leads to failures in software projects [8]. Hence, it is important to understand the situations that lead to biases and avoid them. This will help in the growth of software industry. This paper aims to study the various mechanisms that lead to biases, by analyzing various features of the developers and propose a methodology that solves the problem of bias in software estimation. Section2 gives a brief description about the software estimation biases. Section 3 discusses the experiments conducted and section 4 gives the result of the experiments. Section 5 is the Conclusion

2. Software Estimation Biases

In [9], Magne Jorgenson and Stein Grimstad made a detailed study to understand the mechanisms that lead to the estimation biases by studying developers belonging to different nations including their culture and thinking style. D.Oyerman and S.W.S.Lee discuss about the cultural influence in thinking style [10]. E.U. Weber and C.H. Hsee conclude that the levels of attention to culture are very low and progress could be made by having cross cultural approach. The characteristic related to culture has the impact on estimation bias. The people who are more independent self-construal would incline towards more socially desirable responses. The people who think more holistically would be more exposed to include irrelevant information. In today's world, a software project is developed by people belonging to different nations or different parts of the same nation because outsourcing has become so common. Hence developers belong to different culture and different backgrounds.

So studying such a varied population helps us to understand how the size of the estimation bias is related with variables that depend on different factors including education, culture, technical skills, etc. Hence our study includes the participants from different backgrounds and with various skill set. However, it is difficult

to generalize the results from a sample of questionnaire. The interpretation of the skills using pre-defined scales is a quite challenging task to perform. The developers are requested to answer a few questions before they started the actual work. We propose a new methodology that analyzes the relationship between the bias in software effort estimation and other information about software developers.

3. Experiment Design

The experiment basically consists of three parts. The Project Leader analyzes the developer's background information and skills. According to the result the estimation bias is studied. This is shown in the framework given below:

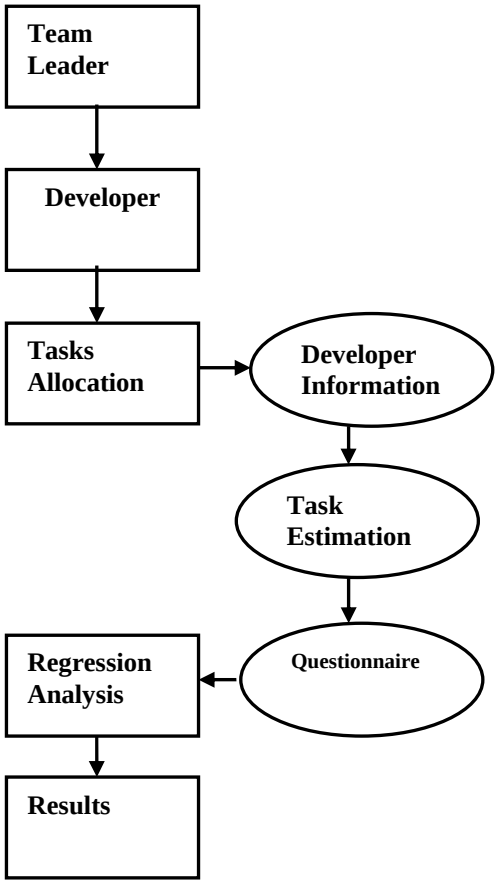

Fig. 1: Software Estimation Bias

In the first part, the background information about the developer is got. This includes the nationality, sex, education, technical skills, experience level, area of specialization, etc. In the second part, task of estimation is done by the software developers. Estimation largely depends on the requirement specifications. The previous studies have clearly shown how misleading and irrelevant information might lead to bias in estimation. The Lines of Code [LOC] is used as an important parameter for the calculation of estimation. The estimation method can be Algorithmic methods like COCOMO or Non-Algorithmic methods like Expert Judgment or Analogy [11].

Estimation of effort is the prediction of the number of hours required by the workers to complete a software project. After the completion of task on estimation, the developers are asked to complete a test based on a questionnaire. This comprises the third part in which questionnaires are given to understand self-construal and thinking style of the developers. Though the aim of the experiment has been validated in many studies, they have some limitations also. In [12], Heine *et al* says that some people are very modest that they respond very low to their own abilities. Also some people report more than what they qualify for. The questions that we used are based on widely used questionnaire presented in [13]. This is used to measure the interdependence and independence characteristics of the individuals. They are not mutually exclusive features as pointed out in recent studies. The increased level of interdependence can lead to awareness of context. This could possibly lead to larger estimation biases and low estimation of effort. Magne Jorgenson *et al* suggested that there will not be strong relationship between estimation bias and other parameters like age, experience, etc [14]. The task specific skill is the only possible exception that could affect the bias in estimation [15].

4. Result

Based on the answers for the questionnaire, the skill set of the people involved in the project is analyzed. The variables and their influences are measured and given in the table below:

Table 1: Self-Construal /Thinking Style Vs Education, Experience and Skill

Variable	Category	Inter-dependence	Thinking Style
Education	Bachelor Degree	4.6	4.9
	Master Degree	4.2	4.6
Experience	High	3.5	3.8
	Low	2.5	4.2
Skill	Average	3.8	4.0
	Good	3.6	4.1

The following figures show the distribution of the values and their difference in the form of graphs.

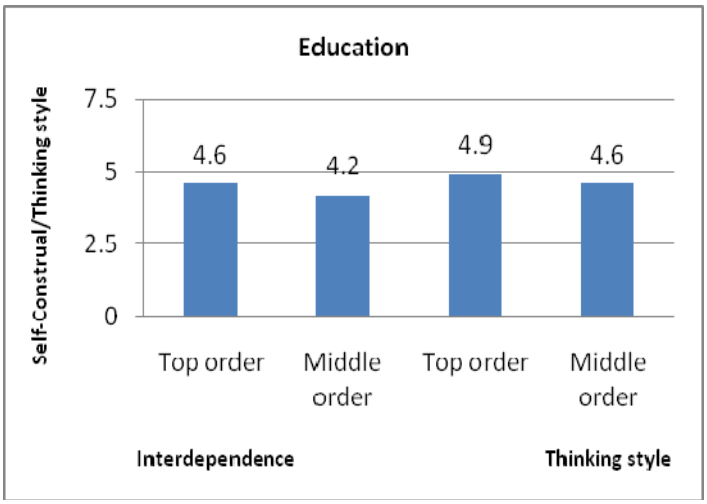

Fig 2: Education Vs Thinking Style

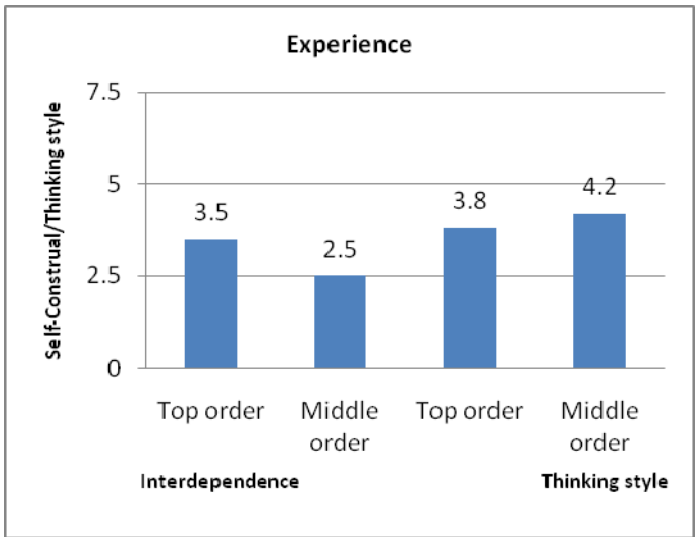

Fig 3: Experience Vs Thinking Style

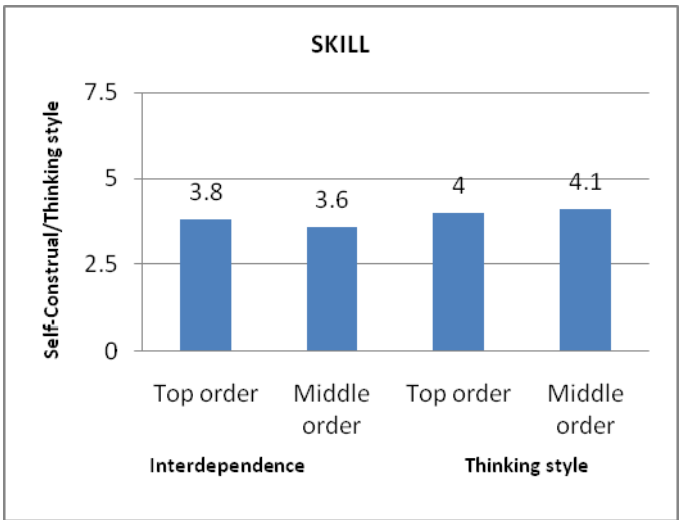

Fig 4 : Skill Vs Thinking Style

To simplify the analysis, the developers in the project are divided into two categories as the Top order Group and Middle Order Group. This division is based on their scores on self- construal and thinking style. The developers with less than 50 percent score are included in the "Middle Order Group". Those developers with 50 percent and more score are included in the "Top Order Group". Regression Analysis is made which suggest that the interdependence has the impact on the estimation. The results clearly indicate the difference between the values. The difference with less than 0.1 suggests that the developers who have more experience is slightly lower in inter-dependence and that those who have higher education are lower on inter-dependence. It can be seen that developers score high in inter-dependence. Also it is found that the effort estimation is also lower in this case. This is because highly developed interdependent self-construal individual look for indirect messages in communication. Hence they are easily affected by misleading and irrelevant information in the task of estimation.

When the attention to the context is more, it could be an advantage and a disadvantage. If the context is relevant, it can be an advantage and it can be a disadvantage if it contains irrelevant information. This relationship is also explained by Krishna *et al* in [16]. B. Hannover *et al* also give similar views on connection between the motivated reasoning and interdependence [17]. Hence we should avoid irrelevant and misleading information when estimating effort. Some methods and principles on this are given by M.Jorgenson in [18].

5. Conclusion

Bias in the estimation of effort is the unavoidable. There are culturally related variables which are used to understand the bias in the estimation. In this paper, it is shown how various parameters like education, skills, experience, etc has the influence over the software effort estimation. Also the analysis and results clearly depicts how an increased level of interdependence is related with the increase in estimation bias. This knowledge is very useful for software clients who develop software while estimating the project. This also helps in planning the software development. This paper emphasizes the importance to address the issue of misleading and irrelevant information.

References

[1] Linda M Laird. (2006) : The Limitations of Estimation, IT Pro, pp. 40-45.
[2] Ning Nan and Donald E.Harter. (2009) : Impact of Budget and Schedule Pressure on Software Development Cycle time and Effort, IEEE Trans. on Software Engg., pp. 624-637.
[3] M. Jørgensen and D.I.K. Sjøberg. (2001) : Impact of Effort Estimates on Software Project Work, Information and Software Technology,vol. 43, no. 15, pp. 939-948.
[4] Tim Menzies, Andrew Butcher, David Cok, Lucas layman, Forrest Shull, Burak Turhan. (2013): Local vs Global lessons for defect Prediction and Effort Estimation, IEEE Transactions on Software Engg., Vol.39, pp. 822-834.
[5] Ekram kocaguneli , Tim Menzies, Ayse Basar Bener and Jacky W Keung. (2012): Exploiting the essential assumptions of Analogy based Effort Estimation, IEEE Engg., pp. 425-437.
[6] Magne Jorgensen and Stein Grimstad. (2011) : The impact of irrelevant and misleading Info. on Software development Effort Estimates : A Randomized Controlled Field Experiment, IEEE Transactions On Software Engg., pp. 695-707.
[7] E. Pronin. (2007) : Perception and Misperception of Bias in Human Judgment, TRENDS in Cognitive Sciences, vol. 11, no. 1, pp. 37-43.
[8] K. Moløkken and M. Jørgensen. (2003) : A Review of Software Surveys on Software Effort Estimation, Proc. Int'l Symp. Empirical Software Engineering.
[9] Magne Jorgensen and Stein Grimstad. (2012) : Software Development Estimation Biases : The Role of Interdependence, IEEE Transactions On Software Engg., vol 38, No. 3, pp. 677-693.

[10] D. Oyserman and S.W.S. Lee. (2012) : Does Culture Influence What and How We Think? Effects of Priming Individualism and Collectivism, Psychological Bull., vol. 134, no. 2, pp. 311-342.
[11] I.Thamarai, Dr.S. Murugavalli. (2012): Using Differential Evolution in the Prediction of Software Effort, Proc. of Fourth International Conference on Advanced Computing, pp. 1-3.
[12] S.J. Heine, D.R. Lehman, K. Peng, and J. Greenholtz. (2002): "What's Wrong with Cross-Cultural Comparisons of Subjective Likert Scales? The Reference-Group Effect," J. Personality and Social Psychology, vol. 82, no. 6, pp. 903-918.
[13] T.M. Singelis. (1994) : The Measurement of Independent and Interdependent Self-Construals, Personality and Social Psychology Bull., vol. 20, no. 5, pp. 580-591.
[14] M. Jørgensen, B. Faugli, and T. Gruschke. (2007) : Characteristics of Software Engineers with Optimistic Predictions, J. Systems andSoftware, vol. 80, no. 9, pp. 1472-1482.
[15] M. Jørgensen and S. Grimstad. (2008) : Avoiding Irrelevant and Misleading Information When Estimating Development Effort, IEEE Software, Vol. 25, No. 3, pp.78-83
[16] A. Krishna, R. Zhou, and S. Zhang. (2008) : The Effect of Self-Construal on Spatial Judgments, J. Consumer Research, vol. 35, pp. 337-348.
[17] B. Hannover, C. Pohlmann, and A. Springer. (2005) : Implications of Independent versus Independent Self-Knowledge for Motivated Social Cognition: The Semantic Procedural Interface Model of the Self, Self and Identity, vol. 4, pp. 159-175.
[18] M. Jørgensen. (2009) : How to Avoid Selecting Providers with Bids Based on Over-Optimistic Cost Estimates, IEEE Software, vol. 26, no. 3, pp. 79-84.