AN EXPERIMENTAL STUDY ON OBSTACLES AND MANAGEMENT ISSUES IN SOFTWARE COST ESTIMATION

[1]VAHID KHATIBI BARDSIRI, [2]AMIN MORADBEIKY, [3]ELHAM KHATIBI

[1,2,3]Department of Computer Engineering, Bardsir Branch, Islamic Azad University, Bardsir, Iran
E-mail: kvahid2@live.utm.my

Abstract- The success of software projects strongly depends on accurate estimation of development effort. Pessimistic or optimistic estimates are the main reasons behind the failure of software projects. Different nature of software projects makes the process of effort estimation very complicated. Despite the fact that numerous effort estimation methods have been proposed, the accuracy of estimates is not convincing and the attempts continue to improve the performance of existing estimation methods. Prior studies have mostly focused on numerical and quantitative approaches to increase the accuracy of estimates. Since the software projects are entirely different from other project types, the lack of analytical and survey-based studies can lead to unreliable and inaccurate estimates inside the organizations. The managers must be aware of the situation of organization in terms of effort estimation. This paper proposes an evaluation framework to measure and evaluate the key indicators of effort estimation inside the organization. The indicators include limitations, obstacles and management issues. A survey was conducted by participation of 135 software developers of a large software company. The experimental results were analyzed and discussed to clarify the status of indicators inside the organization. Since the capabilities and shortages for effort estimation are not the same among the organizations, the proposed framework can be utilized to show the real situation of any organizations in the field of software development effort estimation.

Keywords- Software projects, Cost estimation, Survey, Obstacles, Management Issue.

I. INTRODUCTION

Project management is one of the most important activities performed throughout the software projects. Main phases of project including analysis, design, implementation and deployment are entirely dependent on project management process. All policies, milestones and responsibilities are organized in project management plan. It is undeniable that planning and scheduling of project is a critical part of project management regardless of project type. In first steps of project, project management team should decide on several important questions related to project planning such as how to arrange development team, how to distribute the responsibilities, how to determine deadline for artifacts, how to determine the duration of project and so on. Appropriate response to these questions can ensure the success of software project. On the other hand, careless answering and lack of attention to planning aspects of project may lead to project fault. Knowledge of project management team regarding the project attributes has a considerable effect on dealing with the mentioned questions. Development effort is a key attribute of project that influences on most planning and managing aspects. This attribute refers to amount of effort required for project development.

It comprises of all activities done within different phases of project. Development effort is basis of decision making on management issues at first steps of project. Accurate forecasting the amount of effort required for performing the project will make the development process so smooth and convenient. This is why so many researchers have tried to increase the accuracy of software development effort prediction using various techniques.

Software projects are strongly different than other projects because the purpose of software projects is producing an intangible product [1]. This fact makes the production cycle to be so complicated and difficult in software projects. Therefore, complexity level of software project management is more than other projects. Software project managers are confronted with uncertain and unstable production which is hard to control. Moreover, customer requirements, development technologies and tools are changing rapidly in this field. All of these make the prediction of development effort to be difficult in software projects. As a solution, analyzing of effective factors on development effort estimation can alleviate the problems existing in this area. Investigation of project attributes, limitations, management issues and knowledge of developers in this area can be useful to draw a conclusion in terms of effective factors on management of effort estimation in software projects.

II. STUDY BACKGROUND

In 1973, Interactive productivity and Quality (IPQ)[2] was proposed by IBM group as the first automated tool for software development effort prediction. Afterward, Constructive COst Model (COCOMO) was invented by Barry Boehm [3]. COCOMO utilizes some effort drivers to forecast the amount of development effort. It offers several equations based on complexity level of project. "Software Engineering Economics" [3] is a famous book in this area that still numerous researchers employ proposed

models in which for effort prediction. Putnam Lifecycle Management (SLIM) [4]and Software Evaluation and Estimation of Resources – Software Estimating Model (SEER-SEM) (Galorath Inc.,1980) have used similar principals to COCOMO [5]. In all mentioned models, Line of Code (LOC) was used for designing the prediction model. In fact, development effort was predicted using LOC as size of project.

Function Point (FP) is so important sizing parameter proposed by Albrecht [6]. It was the first idea for measuring the size of software project by using a functional method. Using of FP showed that it can be placed in effort prediction models instead of LOC because computing process of FP is more reliable and accurate than LOC. Advantages of FP motivated researchers to invent new prediction models based on function point such as Albrecht-Gaffney[6], Kemerer [7] and Matson, Barrett and Mellichamp[8]. Introducing of the new version of COCOMO namely COCOMO II in 2000 [9] is a significant event in this field. COCOMO II considers more details of software project for effort prediction. Prediction equations in this method were improved by applying several scale factors.

In contrary to static methods, there are several dynamic models which rely on using past projects information. Classification And Regression Tree (CART) [10] is one of the dynamic methods in this area. It makes a regression tree according to the available information of completed projects and uses the tree to predict the effort of new project. Analogy Based Estimation (ABE) is the other dynamic method proposed in 1997 [11]. ABE method works based on comparing the attributes of new project and past projects to predict the development effort. It is still so popular because it follows simple and straightforward methods for prediction. ABE have been used widely in recent years [12-15]. Latest advancements in prediction of development effort are related to using of soft computing techniques. Neural networks [12, 16] and fuzzy techniques [13, 17-19] are most important soft computing methods employed in this field.

III. PRIOR SURVEY-BASED STUDIES

Several studies [20-25] have investigated the accuracy of schedule and effort estimation, which the results showed that 59%-76% of projects exceeded the determined effort and 35%-80% of which exceeded the determined time. Mean value has been utilized in most previous studies to sum up the overruns in time and effort.

Exceeding the effort indicated in range of 18% and 41%, while the overrun in time is stated in range of 22% and 25% [20-21, 24-25]. According to the latest Chaos report of Standish group, 32% of software projects are successful, 24% fail and 44% are in challenge.

Since project managers may take small cost and effort overruns [26] easy, it can be helpful to realize the status of effort overruns and recognize the projects which involved in significant effort and cost overruns. Moløkken-Østvold [21] used figures to explain the status of effort overruns which results stated that high number of projects exceeded the determined estimates (below 21%) but only a few projects exceeded effort by higher than 100%. Totally, from this research, it can be said that the mean exceeding in effort (44%) was higher than the median of which (21%). Moløkken-Østvold realized that large projects were more intended to be under estimated. They also investigated if the size of project influenced the accuracy of estimates. It must be said that due to limited size of sample, it was difficult to rely on conclusions from statistical aspect.

Previous surveys [21-22, 27-28] have reported that most projects utilized expert judgment or analogy to estimate the effort while only 14%-26% of which utilized algorithmic estimation techniques. The algorithmic techniques comprise of common models such as, COCOMO, Use-Case models, FP-based models and so on.

Several researches attempted to find the cause for the low acceptance level of algorithmic techniques. For instance, most of algorithmic methods are unable to present enough reliable and accurate estimates [29], many companies do not gather enough data to allow the development of algorithmic models [30], organizations and companies feel not well to utilize techniques that they are unable for fully understand [31] and others.

Prior researches investigated the significance of effort estimation and they achieved approximately the same conclusions. Lederer [23] indicated that almost 84% of the developers ranked effort estimation as "very important" or "moderately important". On the other hand, Moløkken-Østvold [32] indicated that 78% of the respondents rated estimation as "most important", "very important" or "extremely important".

Investigating whether organizations and companies have accepted the existing software effort estimation methods is a critical issue in this field. If they are satisfied, they will have no decision to enhance the estimation methods. Otherwise, they can pay more attention toward its improvement. However, it is not as direct and simple as that. Lederer [23] found that, even though development effort estimation is important, developers are neither specially agree nor disagree with the existing methods. The mean rank was 3.02 on one-to-five point scale (1=very disagree 5=very agree). The author indicated that, in terms of

the considerable importance of effort estimation and existing inaccurate estimates, the acceptance of developers indicates that they are satisfied with current methods and they accept the inaccurate estimates.

Moores and Edward [26] indicated that 91% of the responding managers and developers said 'yes' to answer the question 'do you see estimation as a problem?', while only 9% answered 'no'. If this is correct, then, it is true that developers and managers have accepted this problem as a fact of project.

As development phases proceed, the knowledge of developers for software effort estimation becomes more and more, and various estimation techniques are applied at different project stages for any organizations.
It is explained in [3] that the uncertainty existing in effort estimates shows a decreasing process as the project proceeds, which is called the Cone of Uncertainty [33] . As an addition, Gryphon stated that the amount of Uncertainty cannot be decreased automatically, but it can be decreased by the accurate estimation techniques as the development phase progresses [34].

This matter is addressed by Lederer [35] and found that 77% projects performed estimation during the primary stages of project, 64% projects performed estimation at feasibility study phase, while 51% within requirements analysis and 48% in requirement design.

However, the software project aspects and process have changed significantly since the early 1990's, where the survey by Lederer was conducted.

IV. EVALUATION FRAMEWORK

Development effort is one the most important software metrics that must be estimated as accurate as possible. This metric considerably affects the important parts of software projects including planning, scheduling and managing. Lack of paying attention to this metric may leads the project to unpredictable situations. Complexity and uncertainty of software projects causes difficulties confronted by former estimation models.

This is because the current estimation models rely on soft computing and mathematical techniques that are unable to overcome the non-normality of software projects. The role of developers in effort estimation process is significantly important because most of estimation indicators are strongly related to developers' activities. This role can be understood and analyzed only by analytical and survey-based studies. This papers aims to conduct a survey in which three main aspects of effort estimation comprises of management issues, limitations and obstacle are investigated.

4.1 Management Issues

Improvement of effort prediction needs to consider several management practices. Effort prediction is an iterative activity and it requires controlling and managing. Careful management and control of effort predict process can ensure the accurate estimates. On the other hand, lack of management practices may leads the project towards fail. In this section, several management and control practices are represented to be investigated through survey. We are going to explore the importance level of these practices for effort prediction. Moreover, it is investigated that how organizations and companies consider the management practices in effort prediction. Table 1 shows the management and control practices investigated in this study. As seen in the table, believe, attention, documentation, team work, management of uncertainty, and monitoring are the main the sections to be evaluated.

Table 1. Management issues related to estimation which must be investigated

Believe	Estimate the most likely software costs. 1) Do not be confused with the bidding strategy type, political offer. 2) Do not be confused with the software project planning process, such as: the project manager may be in order to improve the team of urgency to make the plan cost less than the cost of the most likely. Managers are aware of benefits of accurate effort estimation Managers do not think that effort estimation is impossible
Attention	Clearly define activities for effort estimation Allocate staff to conduct the effort estimation Define milestone and plan for effort estimation Continues training of developers in field of latest effort estimation tools and methods
Documentation	Clearly defined and documented cost estimates, methods and techniques used, assumptions, and the participants.
	Create a database of historical project effort factors. Measure of the impact of recorded factors in the history on the project effort (such as: software size, software complexity, demand variability, the novelty of the technology, the programmer

	experience level ...), as a basis for effort estimation and control.
Team work	The team recognized. Estimation process and the results are obtained through consultation, the project team recognized.
	Task split. Consider how appropriate split in order to improve the estimation accuracy of the software system or task.
	Use and refer to a variety of methods at the same time. Refer to a variety of cost estimating methods of estimation results (such as: expert judgment, the category, the benchmark comparison, specification estimation model) to compare similarities and differences and to improve the accuracy of the estimates.
Management of Uncertainty	Uncertainty and risk factor analysis. Analysis and determine the possible cause estimates inaccurate and out-of-control. Such as: unstable demand, change the development process, lack of historical project information as a basis for estimation, the lack of monitoring of the cost.
	Estimating uncertainties. Taking into account the estimated uncertainties, given the scope of the effort may be floating.
	Reduces uncertainty. The planned project activities as soon as possible to reduce the cost of uncertainty, to manage the associated risks.
Monitoring	Timely adjustments to estimate the target. According to the software project's progress, the estimated effort is adjusted to achieve the required accuracy.
	Assessment of estimates. Effort estimates are evaluated by an independent person.
	Continuous improvement. Accurately record effort estimates and the effect of cost control, continuous improvement effort estimation and management.

4.3 Specifications of Respondents

There are 253 developers located in three sections of governmental, private and educational. Based on the collected questionnaires, a total of 135 developers have participated in this survey. The number of developers in three sections is 58, 33 and 44, respectively.The developers have different level of skill and experiences. This is the reason why this parameter has been investigated through the questionnaire. The range of developers experience is displayed in Table 2. As seen in the table, most of developers have more than four years of experience. The average of experience for developers is six years, which increases the validity of responses.

Table 2. Experience of developers

Experience (year)	Frequency	Percentage
2.00	1	.7
4.00	4	3.0
5.00	65	48.1
6.00	20	14.8
7.00	23	17.0
8.00	15	11.1
9.00	5	3.7
10.00	2	1.5
Sum	135	100.0

Any project in the selected company (also in most of software companies) is carried out by a specified team. The developers who answered to the questions belong to the different teams. Each team has its own responsibilities and size according to the projects is allocated to do. It was observed that the most percentage of developers belongs to the teams having size of 21 to 40 persons. More than 60 developers are working in the teams with size of 21 to 40 persons. In addition, there are a few developers working in the teams having size of 1-5 or 61 to 80 persons. Although most of developers belong to the relatively large teams, the range of team size is wide enough to investigate the effort estimation aspects in software projects.

V. EXPERIMENTAL RESULTS

5.1 Limitations and obstacles

Therefore, the developers have been asked to determine the importance level of each obstacle in a range of very low to very high. This can show the overall limitations that an specific organization is dealing with. The outcome of developers' responses to each obstacle has been summarized in Table 3. For each obstacle, it is clear that how developers selected an importance level (range of very low to very high).

5.2 Management of effort estimation

Table 4 shows the results obtained from the opinions of developers in terms of management issues. The results were presented based on the fifteen indicators defined to evaluate the management issues. Fifteen indicators have been considered in this study to measure the status of software development effort estimation inside the organizations. These indicators were classified into six groups of believe, attention, documentation, teamwork, management of uncertainty and monitoring. This measurement can be helpful for developers and managers to know the real

situation of organizations in terms of effort estimation. Table 4 displays the responses of developers to management indicators. As seen in the table, it is clear that how the status of each indicator is. For instance, 53.3% of developers have performed the first indicator while the rest (46.7%) have not performed it.

Table 3. The effect of obstacles on accuracy of estimates

Obstacle	Very low		Low		General		High		Very High		Sum	
	Freq.	%	Freq.	%	Freq.	%	Freq.	%	Freq.	%	Freq.	%
Frequent changes in a software requirements	3	2.2	12	8.9	24	17.8	66	48.9	30	22.2	135	100.0
The software requirements are not clear	10	7.4	19	14.1	23	17.0	53	39.3	30	22.2	135	100.0
Lack of historical project data	12	8.9	21	15.6	26	19.3	58	43.0	18	13.3	135	100.0
Lack of appropriate estimation methods or estimation process	11	8.1	18	13.3	10	7.4	71	52.6	25	18.5	135	100.0
Lack of the support of the estimation tools	7	5.2	13	9.6	11	8.1	52	38.5	52	38.5	135	100.0
Not enough time or manpower to carry out the effort estimate	16	11.9	44	32.6	8	5.9	46	34.1	21	15.6	135	100
Pressure from senior managers, customers, or others, directly specify or modify the estimation results	4	3.0	28	20.7	29	21.5	55	40.7	19	14.1	135	100
Lack of participation of application developers	5	3.7	19	14.1	16	11.9	69	51.1	26	19.3	135	100
Lack of timely supervision and control of cost according to plan	9	6.7	26	19.3	30	22.2	57	42.2	13	9.6	135	100
Lack of analysis of software systems and the associated risks	21	15.6	30	22.2	21	15.6	36	26.7	27	20.0	135	100
Lack of coordination among the relevant stakeholders of the customers, users, system design and development,	20	14.8	35	25.9	33	24.4	36	26.7	11	8.1	135	100
Lack of risk analysis and management of software projects	10	7.4	19	14.1	16	11.9	61	45.2	29	21.5	135	100
Is difficult to estimate the skill level of the project members in technologies	2	1.5	17	12.6	14	10.4	57	42.2	45	33.3	135	100

Table 4. Management issues

Indicator	Done		Not-Done		Sum	
	Freq.	%	Freq.	%	Freq.	%
Estimate the most likely software costs. 1) Do not be confused with the bidding strategy type, political offer. 2) Do not be confused with the software project planning process, such as: the project manager may be in order to improve the team of urgency to make the plan cost less than the cost of the most likely.	72	53.3	63	46.7	135	100.0
Effort estimates based on reliable input. Such as: the basis for software size, history project experience.	55	40.7	80	59.3	135	100.0
Documented. Clearly defined and documented Effort estimates, methods and techniques used, assumptions, and the participants.	55	40.7	80	59.3	135	100.0
Trade-offs and adjustments. When estimates derived from the most likely software costs, trade-off between the elements of cost, schedule, functionality, quality, technology and process, and update the estimates based on adjusted accordingly.	48	35.6	87	64.4	135	100.0
The team recognized. Estimation process and the results are obtained through consultation, the project team recognized.	61	45.2	74	54.8	135	100.0
Task split. Consider how appropriate split in order to improve the estimation accuracy of the software system or task.	72	53.3	63	46.7	135	100.0
Process monitoring. Monitor the implementation of the software project plan, the timely detection of deviation from the project plan.	74	54.8	61	54.12	135	100.0
Uncertainty and risk factor analysis. Analysis and determine the possible cause estimates inaccurate and out-of-control. Such as: unstable demand, change the development process, lack of historical project information as a basis for estimation, the lack of monitoring of the cost ...	69	51.1	66	48.9	135	100.0
Estimating uncertainties. Taking into account the estimated uncertainties, given the scope of the effort may be floating.	53	39.3	82	60.7	135	100.0
Reduces uncertainty. The planned project activities as soon as possible to reduce the cost of uncertainty, to manage the associated risks.	56	41.5	79	58.5	135	100.0
Timely adjustments to estimate the target. According to the software project's progress, the estimated effort is adjusted to achieve the required accuracy.	77	57.0	58	43.0	135	100.0
Assessment of estimates. Effort estimates are evaluated by an independent person.	67	49.6	68	50.4	135	100.0
Create a database of historical project effort factors. Measure of the impact of recorded factors in the history on the project effort (such as: software size, software complexity, demand variability, the novelty of the technology, the programmer experience level ...), as a basis for effort estimation and control.	80	59.3	55	40.7	135	100.0
Use and refer to a variety of methods. Refer to a variety of cost estimating methods of estimation results (such as: expert judgment, the category, the benchmark comparison, specification estimation model) to compare similarities and differences and to improve the accuracy of the estimates.	96	71.1	39	28.9	135	100.0
Continuous improvement. Accurately record effort estimates and the effect of cost control, continuous improvement effort estimation and management.	75	55.6	60	44.4	135	100.0

DISCUSSION AND CONCLUSION

In this paper, a survey was conducted to investigate the real situation of effort estimation inside the organizations. A total of 135 developers participated in this survey to clarify the critical issues in the field of effort estimation. The limitations, obstacles and management issues were the main aspects investigated through the current survey.

As the results of this survey confirmed, the most important obstacles against accurate effort estimation inside the organization are the lack of support for estimation tools and the difficulty of estimation of members' skill level. Moreover, the lack of coordination has been mentioned as the obstacle that has the lowest level of importance. This can show the appropriate cooperation and collaboration among the developers working inside the organizations. Regarding the management issues, it was observed that the most ignored task is uncertainty estimation inside the organization while referring to a wide range of estimation methods is the most important task.

The method of investigation proposed in this paper can be utilized in any organization types to determine the status of effort estimation issues. Due to the diversity of activities and goals of organizations, the principles of this survey must be considered prior to estimation process. This can be helpful to find an overall scheme for current situation of organization in terms of effort estimation.

The strengths, weaknesses and important factors can be found through a survey and the results can help managers to improve the quality of effort estimation. As the future work, we are going to increase the domain of investigation carried out inside the organization.

REFERENCES

[1] G. Stepanek, Software Project Secrets: Why Software Projects Fail (Apress, USA, 2005).

[2] C. Jones, Estimating software costs: Bringing realism to estimating, 2nd ed. (NY: McGraw-Hill, New York, 2007).

[3] B. W. Boehm, Software engineering economics (NJ: Prentice Hall, Englewood Cliffs, 1981).

[4] L. H. Putnam, A General Empirical Solution to the Macro Software Sizing and Estimating Problem, IEEE Transactions on Software Engineering,, SE-4(4) (1978) pp. 345-361.

[5] B. W. Boehm and R. Valerdi, Achievements and Challenges in Cocomo-Based Software Resource Estimation, IEEE Softw., 25(5) (2008) pp. 74-83.

[6] A. J. Albrecht and J. A. Gaffney, Software function, source lines of codes, and development effort prediction: a software science validation, IEEE Trans Software Eng. SE, 9(6) (1983) pp. 639-648.

[7] C. F. Kemerer, An empirical validation of software cost estimation models, Commun. ACM, 30(5) (1987) pp. 416-429.

[8] J. E. Matson, et al., Software Development Cost Estimation Using Function Points, IEEE Trans. Softw. Eng., 20(4) (1994) pp. 275-287.

[9] B. Boehm, Software Cost Estimation With COCOMO II (Prentice Hall, 2000).

[10] L. Breiman, et al., Classification and Regression Trees (Pacific Grove, CA: Wadsworth, 1984).

[11] M. Shepperd and C. Schofield, Estimating Software Project Effort Using Analogies, IEEE Transaction on software engineering,, 23(11) (1997) pp. 736-743.

[12] Y. F. Li, et al., A study of the non-linear adjustment for analogy based software cost estimation, Empir Software Eng, 14(2009) pp. 603-643.

[13] M. Azzeh, et al., Fuzzy grey relational analysis for software effort estimation, Empirical Software Engineering, 15(1) (2010) pp. 60-90.

[14] Q. Song and M. Shepperd, Predicting software project effort: A grey relational analysis based method, Expert Systems with Applications, 38(6) (2011) pp. 7302-7316.

[15] C.-J. Hsu and C.-Y. Huang, Comparison of weighted grey relational analysis for software effort estimation, Software Quality Journal, 19(1) (2011) pp. 165-200.

[16] C. S. Reddy and K. Raju, A Concise Neural Network Model for Estimating Software Effort, International Journal of Recent Trends in Engineering, 1(1) (2009) pp. 188-193.

[17] P. V. G. D. Prasad Reddy, et al., Fuzzy Based Approach for Predicting Software Development Effort, Software Engineering(IJSE), 1(1) (2010) pp. 1-11.

[18] K. R. Ch. Satyananda Reddy, Improving the accuracy of effort estimation through Fuzzy set combination of size and cost drivers, WSEAS Transactions on Computers,, 8(6) (2009) pp. 926-936.

[19] H. K. V. Vishal Sharma, Optimized Fuzzy Logic Based Framework for Effort Estimation in Software Development, International Journal of Computer Science, 7(2) march 2010 (2010) pp. 30-38.

[20] A. M. Jenkins, et al., Empirical Investigation of Systems Development Practices and Results, Information & Management, 7(1984) pp. 73-82.

[21] K. Moløkken, et al., A survey on software estimation in the Norwegian industry, presented at the 10th International Symposium on Software Metrics, 2004.

[22] F. J. Heemstra, Software cost estimation, Information and Software Technology, 34(10) (1992) pp. 627-639.

[23] A. L. Lederer and J. Prasad, Causes of Inaccurate Software Development Cost Estimates, Journal of Systems and Software, 31(1995) pp. 125-134.

[24] F. Bergeron and J.-Y. St-Arnaud, Estimation of Information Systems Development Efforts: A Pilot Study, Information & Management, 22(1992) pp. 239-254.

[25] C. Sauer and C. Cuthbertson, The State of IT Project Management in the UK 2002-2003, Templeton College, University of Oxford, 2003.

[26] T. T. Moores and J. S. Edwards, Could Large UK Corporations and Computing Companies Use Software Cost Estimating Tools? A Survey, European Journal of Information Systems, 1(5) (1992) pp. 311-319.

[27] G. Wydenbach and J. Paynter, a Survey of Practices in New Zealand Software Project Estimation, Zealand Journal of Computing, 6(1B) (1995) pp. 317-327.

[28] K. McAulay, Information Systems Development and the Changing Role of MIS in the Organisation, presented at the First New Zealand MIS Management Conference, Wellington, 1987.

[29] L. C. Briand , et al., COBRA: A hybrid method for software cost estimation, benchmarking and risk assessment, presented at the Proceedings of 20th International Conference on Software Engineering, 1998.

[30] M. Ruhe, et al., Cost estimation for web applications, presented at the 25th International Conference on Software Engineering, Portland, Oregon, 2003.

[31] M. Jorgensen, A review of studies on expert estimation of software development effort, Systems and Software, 70(2004) pp. 37-60.

[32] K. Molokken and M. Jorgensen, A review of software surveys on software effort estimation, in Empirical Software Engineering, Washington, DC, USA, pp. 223-230, 2003.

[33] S. McConnell, Rapid Development: Taming Wild Software Schedules. (Microsoft Press, 1996).

[34] S. Gryphon, et al., Letters: The Cone of Uncertainty, IEEE Software, 23(5) (2006).

[35] A. L. Lederer and J. Prasad, Nine Management Guidelines for Better Cost Estimating, Communications of the ACM., 35(25) (1992) pp. 51-59.

★★★