Analysis of Combining Software Estimation Techniques

M. V. Deshpande
SVKM's D.J.Sanghvi College of Engineering.
Vile Parle (West),
Mumbai, India.

S. G. Bhirud
Veermata Jijabai Technological Institute,
Matunga,
Mumbai, India

ABSTRACT

In this paper, analysis of the result obtained from experiment with three projects is described. Combination of different estimation techniques helps estimator to reduce error and keep control over the deviation of estimates away from actual. As a response to survey from estimators working in software industry, It is revealed that, according to stages, estimates are termed as budgetary, initial, progressive and closure. As a budgetary estimates are dominated by analogy based estimation techniques, It can be complemented by COCOMO II Application Composition Model. The Initial estimates is average of COCOMO II Early Design model with Object point sizing, which can be complemented by Function point and Usecase point based estimation. The Progressive estimates are calculated by averaging of COCOMO II Post Architecture Model and Class point based estimates. It is observed that effort estimates are more accurate than using only COCOMO II.

Keywords

Software estimation, Usecase point, Function point, Class point, Budgetary, Progressive

1. INTRODUCTION

In this age of automation, estimating a software is an activity which is preferred to be done manually using minimal tools. Automating the estimation process partially if not fully became a challenge and an inspiration to conduct the experiment[1]. Moreover, capturing expert knowledge in some form or the other to assist in the estimation activity was a motivating factor that helped in designing techniques that can automate some of the methods used for estimation. This needed a lot of study, survey of current industrial trends and research in the field of estimation.

The survey has been conducted as initial activity of the experiment[2]. It is revealed from the responses to the questionnaire that estimation has prime importance to take managerial, commercial, enterprise-wide decisions. The question related to deviation of estimates from the actual, was responded by all professional estimators that at initial stage +25 % to -25 % can be acceptable if there is a contingency[5]. At the end of Design it should be +10% to -10 % and at the end of project, deviation must be within 5% at higher or lower side of actual effort spent for the project[9]. Another view of these responses is that resources are reserved according to the initial estimates and these resources are adjusted as project progresses but the range must be within at lower and higher limit of previous resource reserves. Keeping these views in the mind experiment has been conducted to calculate estimates by more than one method at each stage and results are analyzed.

2. BACKGROUND

The sizing of software is very important for effort, schedule and cost estimation. The size of software is represented in terms of KLOC (One Thousand Line of Code), Function point, Object point, Usecase point, Class point. Estimation techniques or model use these sizing parameter as input[3][7]. COCOMO needed KLOC as size input. COCOMO II need Object point or Function point for Application Composition Model. There are estimation methods which accept size as input in the form of Usecase point or Class point[4][6].

Various estimation techniques have been developed by researchers and they have been converted into commercial or freeware tools.

The Rational Unified Process is followed widely as process model for software development. COCOMO II literature suggested applicability of different COCOMO II models during phases of RUP and Software Development Life Cycle[2].

3. THE EXPERIMENT

Three projects viz. P1, P2 and P3 have been selected for the experiment, for which sufficient documentation was available. Initially Request for Proposal document is referred. Each RFP was evaluated for completeness of problem statement. Size in terms of Function points was derived for each. Then usecase diagram along with usecase text are studied and usecase point are computed. The analysis and design class diagrams were referred and class points are computed[8].

The experiment has been conducted with the consideration of RUP as a process model. The estimates are calculated during inception, elaboration phases and mapped as budgetary, initial and progressive estimates.

Budgetary estimates are calculated based on input given by experts from industry. They were asked to apply their knowledge and experience and suggest effort in terms of person-months with justification. The estimate with minimum error with all inputs is selected for each project.

Initial estimates are calculated by following Function point method, COCOMO II Early Design Model and Usecase point method. Each Usecase point is weighed appropriately to calculate effort in Person-Months.

Progressive estimates are calculated by following COCOMO II Post Architecture Model and Class point method.

4. RESULT AND FINDINGS

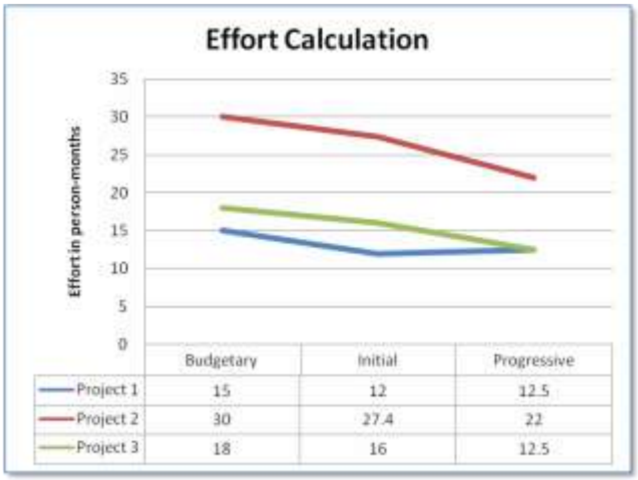
The estimated effort in person-months for three projects are depicted in Table 1.

Project- **Budgetary Estimate**	**Initial Estimate**			**Progressive Estimate**	
	Function Point	**Early Design**	**Use Case**	**Post Architecture**	**Class Point**

			Point		
P1-15	11.6	13.1	11.3	14	10.73
	12			12.5	
P2-30	25.6	29.8	26.7	34.5	10.73
	27.4			22	
P3-18	13.7	17.4	17.3	20	5.28
	16			12.5	

Table 1. Summary of Estimates by different methods

Three projects are selected and estimates are calculated at three stages. For Initial estimates Function point, COCOMO II Early design and Usecase points are averaged. Progressive estimates are average of COCOMO II Post Architecture Model and Class Point Method is applied. These values are plotted in a graph as shown in Figure 1 and trend is observed.



5. CONCLUSION

Figures in the Table 1. are indicative that estimates by more than one method at different stages of project life cycle, helped to converge around actual effort needed to spend for overall project. Since actual effort needed to spend will not be known until end of project, estimates derived from single method cannot be relied[9]. Only COCOMO II can be called as more comprehensive because Cost Drivers reflects various complexity factors. The literature also provides calibrated values of these factors.

Usecase point and Class point based estimation are relatively new methods which need further evaluation to rely solely, but these method can be used as complementary to established estimation methods like COCOMO II.

For any decision related to resource allocation, bidding cost and defining schedule, experience and skill level of human estimator are dominating factors for improving accuracy of estimates.

6. FUTURE SCOPE

This process of estimation can be matured by applying combined approach with different estimation techniques for more projects. The tool can be developed which support estimation at all stages with available project knowledge[1]. This tool can be a integrated component of CASE tool. When RFP document is available, estimation tool should be able to identify similar project completed in past and provide data needed to estimate current project in hand. When requirements are frozen, Function point method should be applied. The tool with Artificial Intelligence features, should be able to read usecases and class diagram converted in XML form to derive usecase point and class point respectively. Finally tool should be intelligent enough to propose estimate at any stage by appropriate statistical processing over estimates derived from different estimation methods[9].

7. ACKNOWLEDGMENTS

Our special thanks to Mr. Pinaki Bhowmik and industry experts for their contribution towards the conduction of survey. The responses to the questionnaire were so useful. We also express gratitude towards Mrs. Kiran Bhowmik for her involvement in this experiment.

8. REFERENCES

[1] Magne Jorgensen, “Realism in Assessment of Effort Estimation Uncertainty: It Matters How you ask”, IEEE Transaction on Software Engineering, vol. 30, no. 4, 2004

[2] Magne Jorgensen, Sinan Tanilkan, Hans Gallis, Annet Lein, Siw Hove Kjetil Molloken-Ostvold, “A Survey of Software Estimation in Norwegian Industry”. IEEE Computer Society. 10th International Symposium on Software Metrics, METRICS'04. 2004

[3] John Smith, Rational Software, “The Estimation of Effort Based on Use Cases”, Rational Software white paper.

[4] Shinji Kusumoto, Fumikazu Matukawa, Katsuro Inoue, Shigeo Hanabusa, Yuuske Maegawa, “Effort Estimation Tool Based on Use Case Points Method”, Osaka University.

[5] Barry Boehm, “COCOMO II Model Definition Manual”, University of Southern California, 2000.

[6] SangEun Kim, William Lively, Dick Simmons, “An Effort by UML Points in the Early Stage of Software Development”, Texas University, USA.

[7] Gennaro Costagliola, Filomena Ferruci, Geoveffa Tortora, Giuliana Vitiello, “Class Point: An Approach for the Size Estimation of Object-Oriented Systems”, IEEE Transactions on Software Engineering, Vol 31, No. 1, Jan. 2005

[8] Bhowmick Kiran, Deshpande M.V., Bhirud S.G., “A Software Estimation combining Algorthmic and Non-Algorithmic Models”, Proc. of National Conference on Information & Communication Technology, NCICT 08, Mumbai, India, 2008.

[9] Bhowmick Kiran, “Integration of AI models with empirical models for software estimation”, M.Tech. Thesis. NMIMS university, 2008.