



Amalgamation of Estimation Techniques in Assessing Software Metrics and their Utilization in Project Management

Swati Chaudhary¹, Madan Singh²

Department of Computer Science Engineering, SRM University(India)

*Corresponding Author Email: ¹swati.chaudhary031@gmail.com, ²madan.phdce@gmail.com

Abstract

The main reason for software-intensive acquisition programs is the poor size estimation that is not more effective. The estimation of the significant software metrics, utilization of the metrics, recording the metrics are the challenges in software estimation. The software estimation becomes a tedious task due to these challenges, which needs to be rectified. In literature, a number of techniques were proposed to resolve these challenges by the researcher. In our paper, we present a survey on the existing software estimation techniques based on the existing issues.

Keywords: Amalgamation, Software metrics, Project Management, Software effort estimation.

1. Introduction

The most of existing software-intensive acquisition programs fails due to the poor size estimation. The size of these programs mainly depends on the cost, schedule, and efforts required to develop these programs. The failure to accurately predict (generally too small) results in the over budget and late deployment which may lose the client support for your program. Any human-intensive activity, without control, deteriorates over time. If we do not measure, there is no way to know whether processes are on track or if they are improving. It takes a fix attention and manage to keep these software-intensive acquisition and the development processes from breaking down and let alone to improved them. Measurement, provides a technique for us to estimate the status of our program to determine if it is in trouble or in need of corrective action and process improvement. Measurement provides benefits at the program, strategic and technical levels. A perfect measurement program is an good investment in growth by checking early detection of problems arising, and by giving quantitative clearness of severe development problems. Metrics give us the ability to identify, clarify, resolve, and curtail certain risk issues before they find. Software estimation tools are an integral part of the IT organizations and an extremely vital phase within the software development life cycle. The significance of estimating factors like Effort, Time & People required in a particular project is so much that they sometimes become essential to win that project from the client. It becomes mostly important in those scenarios where a client is very particular about the time & cost details of a project. Nowadays, the estimation tools not only help in the calculation of estimation factors but also contribute in highlighting their usage in various other vital phases of software development like Project Scheduling, Personnel Planning etc. They help an organization to overcome the challenges involved in planning the schedule,

costing & manpower resources by utilizing the estimation factors appropriately and hence are extremely powerful in giving near-

accurate projections. As we need to calculate the weight, volume, and dynamic flight(example) characteristics of an aircraft as part of the planning phase, we need to check how much software required to build an aircraft..

The reasons behind why software programs fail is our inability to accurately estimate software size. As we always estimate size too low for software projects, we do not fund or allow enough time for development. Poor size estimates are one of the main reasons and usually at the heart of cost and budget overruns. We have chosen to implement this concept considering the significance of software metrics such as Effort, Time & People involved in the development lifecycle of a certain software project/product. With the implementation of this concept, we intend to understand how the specific input parameters about the features of any software application helps in the estimation of its vital factors and tell us about their usage in planning additional phases.

2. Existing Cost Estimation and Project Management Techniques

The reasons behind why software programs fail is our inability to accurately estimate software size. As we always estimate size too low for software projects, we do not fund or allow enough time for development. Poor size estimates are one of the main reasons and usually at the heart of cost and budget overruns. Before developing this application, we considered below research papers:-

According to [1] No model can estimate the high degree of accuracy cost of software. This phase of the practice is created because:

A. There are a large number of interrelated factors that affect the development process of a software given to the development team and a large number of project attributes, e.g., number of user screens, volatility of system requirements and the use of reusable software components. B. The development environment is generating continuously. C. The lack of evaluation that truly denotes the complexity of a software system or a program.

To generate a better estimate which reflects the complexity of a software system, we must improve our understanding of these project attributes as mentioned before and their casual relationships between them, model the impact of generated environment and develop the effective ways of measuring the complexity of a software system.

As we have studied more about the project during Analysis and Design phases, the uncertainties which occur are reduced and more accurate estimates can be made during these phases. Many models produce exact results without noticing this uncertainty.

Whereas [2] presented a method for Cost estimation, Benchmarking, and Risk Assessment: COBRA. This method represents to be convenient and at low cost when an organization needs to develop local estimated cost and risk models and is not able to fetch or collect a large set of project data. For example, the case study shown in this research took approximately 2 man-months of interviewing and their analysis effort. They have also shown how project manager expertise can be collected, refined, modelled, validated, and used for cost overhead estimation and cost risk benchmarking and assessment. Their resulting cost estimation and risk assessment models are operational and their construction is repeatable through a well-defined process. Their case study has shown good initial results on actual projects, thus defining the feasibility of such an approach.

According to [3] this paper contributed a knowledge-based concept to the organisation of selecting appropriate effort estimation methods for software project management. However, the answers to our questionnaire highlighted that such a recommendation application might be used in practice of such approaches. This holds particularly for a complex knowledge base as a recommendation system might have genuine value for entering even more detailed facts about the project under consideration. In the near future they plan to extend the knowledge-base concept to cover more project details.

According to [4] this paper presented effort estimation technically, the usage of Lines of Codes, Function point. Project management is composed of various different types of activities such as, planning the work, estimating resources, allocating of resources, organizing the work, acquiring human and material resources.

According to [5] focused on developing software cost estimation models and techniques. These include algorithmic models, such as

COCOMO, SLIM, SEER-SEM. This gives the idea of cost estimation for any software using effort, people, time etc. But to determine the cost analysis we need SEER-SEM model, because without Effort we can't find out the cost. The main techniques adopted for effort estimation are briefly introduced in this article; these models are classified as experience-based, learning-oriented, model-based, regression-based and composite techniques. Thus we can evaluate the prediction performance of the proposed neuro-fuzzy model with SEER-SEM in software estimation practices.

As per [6] this paper focused on an effort estimation method based on early phase effort.

According to [7] use Hybrid tool Estimator to relate Effort, Time, number of peoples. The unique difference between the proposed and existing estimation of effort for the software system development is the level of quality consideration. That is, existing estimations are using only few quality factors for effort estimation, but the proposed effort estimation covers the important quality factors, which automatically reflects in the development of software.

According to [8] this study has proposed an automated Hybrid tool which calculates Use Case Points from Use Case Models. We will use the effort estimation based on this Hybrid Tool in the hybrid technology proposed for risk assessment and benchmarking.

According to [9] This paper illustrates model with an objective to depict the accuracy and bias variation of an organization's estimates of software testing effort through Cobb-Douglas function (CDF). Using the model the accuracy of effort estimation calculated is 92.42%. It clearly states Cobb-Douglas model is more robust than other models as the model is blend of using historical data and scientific approach.

3. Conclusion

This application is specifically targeted & meant for commercial software organizations that need to estimate, analyse & utilise different vital metrics involved within a software project like Effort, Schedule, Manpower, Complexity, Size, Cost etc. during the project phase & planning phase.

4. Comparison among the Existing Techniques

| UTHORS | PARAMETERS | | | | | | |
|-----------------|------------|------|---------|--------|------|---------|-------------|
| | COST | TIME | EFFORTS | PEOPLE | RISK | TESTING | UTILIZATION |
| Leung et al. | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| Lionel C. et al | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Peischl et al | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| Rao et al | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Capretz et al | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Kamei et al | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Vijay et al | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Manoharan et al | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Umar et al | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ |

References

- [1] Leung H., Fan Z.: Software Cost Estimation The Hong Kong Polytechnic University
- [2] Briand C., Emam K., Bomarius F.: COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. International Software Engineering Research Network Technical Report ISERN-97-24 (Revision 2) Fraunhofer Institute for Experimental Software Engineering (IESE) Germany
- [3] Peischl B., Nica M., Zanker M., Schmid W.: Recommending effort estimation methods for software project management. Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 3 (WPRRS Workshop), Milano, Italy, 2009, pp. 77-80.
- [4] Koteswara K., Raju G.S.V.P., Madhusudhana T.V., Sumendra M., SankarSharath SS.: Effort estimations based on Lines of Code and Function Points in Software Project Management. IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6, June 2008
- [5] Lin Du W., Fernando Capretz L., Bou Nassif A., Ho D.: A Hybrid Intelligent Model for software cost estimation. Journal of Computer Science 9 (11): 1506-1513, 2013 ISSN: 1549-3636.
- [6] Tsunoda M., Kamei Y., Toda K., Nagappan M., Fushida K., Ubayashi N.: Revisiting Software Development Effort Estimation based on early phase development activities.

Proceedings of the 978-1-4673-2935-4/13/\$31.00 c 2013 IEEE
429 MSR 2013, San Francisco, CA, USA.

- [7] Vijay Frank J., Manoharan C.: An Analysis of Hybrid Tool Estimator: An Integration of Risk with Software estimation. Journal of Computer Science 7 (11): 1679-1684, 2011 ISSN 1549-3636 © 2011 Science Publications.
- [8] Vijay Frank J., Manoharan C.: Initial Hybrid Method for Analyzing Software Estimation, Benchmarking and Risk Assessment Using Design of Software. Journal of Computer Science 5 (10): 717-724, 2009 ISSN 1549-3636 © 2009 Science Publications.
- [9] Nafeez Umar S.: Effort estimation with cobb-douglas function: A practical application. IJRET: International Journal of Research in Engineering and Technology ISSN:2319-1163 Volume: 02 Issue: 05 | May-2013