# Factors on Software Effort Estimation models and Techniques – A Detailed Report

Dr.J.Jegathesh Amalraj[1], Dr.D.P.Jeyapalan[2]

Assistant Professor[1], Associate Professor[2]

Department of Computer Science[1&2]

Thiruvalluvar University Constituent College, Thittagudi, Cuddalore[1]

St.Josephs College, Trichy[2]

## Abstract

In the process of software project development life cycle, software effort estimation is the important process. The effort estimation affects the success of software projects whether the designers estimate the projects improperly. Normally, software is grown in size and complexity, it is very difficult to precisely known the cost of software development. The main challenging fact of this effort estimation is the software industry was the fast-changing nature of software development. In this paper assessed the several classes of software effort estimation models and techniques. This paper overviews a research study comparing the different estimation techniques that can be helpful to the information technology firms, researchers and software developers that use information technology in the processes such as budgeting and decision-making processes.

## Keywords

Effort estimation, System Development Life cycle, Software Applications, Software Projects.

## I Introduction

In generally, the software industry incapability provides the accurate estimates of development cost, effort and its time. This incapability is described in reports from project management consulting companies and estimation surveys. Software Effort Estimation (SEE) is the main factor in all successful completion of the project. When project management is a fault it breaks the entire software process [1]. Therefore, it is very essential to learn the various aspects of software project management. The key features of software project management are as follows, i) Project scheduling, ii) Monitoring and Control, iii) Project Estimation, iv) Risk Management and v). Report Generation. Basically, project estimation is the most challenging task because it estimates the size, effort, cost, time and so on. In this above tasks, first of all, we calculate the size of the project, after that, we determine the required effort from the project size and then we calculate the product duration with cost by the estimation of effort [1]. So that, the software size estimation is more important to conclude the project effort. Effort estimation has provoked According to a recent research in 2017, only 30% of the companies skilled in size estimation and 55% expert in software effort estimate [2]. In generally effort estimation is the process of predicting the effort required to developer maintain software product in months. Various ways are available for categorizing effort estimation approaches, the main effective and efficient categories are discussed as follows,

i) Expert Estimation

The quantification step is on the basis of judgmental process estimation is done in this category.

ii) Formal Estimation

The quantification step is based on mechanical processes. For example, the use as a formula is derived from historical data.

iii) Combination-Based estimation

The combination-based estimation approach processed with the combination of judgmental and mechanical combination of estimates from different sources.

## II Literature Review

In 2017, Mohammed Aljohani and Rizwan Qureshi [6] compared a study which considers the accuracy, usage and suitability of the existing methods. The authors said this comparative study is more suitable for the project managers and project consultants during the entire software development process. They discussed that this comparison can facilitate and help the project managers to distinguish and compare among techniques to choose the optimum technique according to project type and requirements. The authors proposed a model which includes the errors that can result due to the single approach failure.

In 2017, Simon WU Iok Kuan [9] proposed a regression model to predict the effort required to design small and medium scale application software. The author developed a model used 60 completed software projects developed by a software company. With the usage of these projects, the author extracted factors and applied them to a regression model.

In 2016, M.Pramod Kumar et al., [8] deals with severe classes of software effort estimation models and techniques. They surveyed about 80 research papers for software effort estimation. The authors discussed authentic estimation of software development effort is a very difficult job. Both under and overestimations can lead to serious consequences. This paper overview a research study comparing the different estimation techniques.

In 2012, P.K. Suri and Pallavi Ranjan [10] summarized a paper for several classes of software cost estimation models and techniques. This assessment framework helps an inexperienced team improve project tracking and estimation.

In 2013, Kjetil Molokken et al., [3] reviewed a survey of software effort estimation. The authors said that this survey also tries to obtain the respondent’s interpretation of estimate.

## III Estimating Efforts

Usually, the project development effort directly depends on the two important inputs namely Application Size and Productivity of programmer [4]. Transforming the size into deliverable efforts within a schedule makes the project complete success. Productivity is based on the programmer's capability and technological adaptability and flexibility in using integrated development environments (IDE's). The main objective of software size estimation is to estimate the effort such as people, schedule, cost needed to develop a software product [4].

For some software projects, simple mistakes may lead to either overestimation or underestimation. The use of excessive resources may affect the software product completion time or schedule [5]. The underestimation development effort can lead to under-staffing, cost and schedule overruns and poor-quality software, while over-estimates can result in inefficiency and wasted resources, as projects tend to expand to fit the estimated effort.

### 3.1 Software Estimation

When the software project is assigned to a project manager, they must know the effort, schedule and functionality in advance [7]. This kind of project factors may change in the duration of a project. The main thing is to predict the factor by which they change. So the process of estimation needs to be carried out. Estimating is the process of forecasting or approximating the time and cost of completing project deliverables or The task of balancing the expectations of stakeholders and the need for control while the project is implemented. The categories of estimation techniques are classified as follows,

**Estimation by Analogy**

Estimating by Analogy means to create estimates for new projects by comparing the new projects to similar projects from the past. As the algorithmic techniques have a disadvantage of the need to calibrate the model [7]. So, the alternative approach is "analogy by estimation". But it requires a considerable amount of computation. This process is much simpler. But not all organizations have historical data to satisfactorily use analogy as a means of estimation. ISBSG (International Software Benchmarking Standards Group) maintains and exploits a repository of International Software Project Metrics to help software and IT business customers with project estimation; risk analysis, productivity, and benchmarking.

**Expert Opinion**

When quantified or empirical data is absent, then expertise based techniques are needed. The opinion of experts is taken, but the drawback with this technique is that the estimate is as good as the expert's opinion only. For example, Delphi technique or work breaks down structure.

## IV Model-based Estimation techniques

Model-based software estimation techniques are the algorithms with mathematical models calibrating from the definite amount of known project data.

The Example of model-based estimation techniques methods [11] is Functional Point Analysis (FPA), Software LIfe cycle Management (SLIM), and COnstructive COst Model (COCOMO) which are essential in planning, tradeoffs analysis, monitoring and estimating resources and fund required for a software project.

### Functional Point Analysis (FPA)

Functional Point Analysis is used for measuring the complexity and size of software. The idea of function points is accessing the size of the system in functions from they perform, and the idea transfers to an object-oriented approach. So that, FPA focuses the use case and class diagram in order to estimate the software project.

### COCOMO Model

Cost Constructive Model (COCOMO) was predicted by Barry W. Boehm in 1981. He gave the following equations to evaluate the effort.

Effort=A × (Size) B

Where,

A =Constants based on different classes of Software Projects

B=Economy or Diseconomy of Scale based on different classes of Software Projects

Size= No. of Source Lines of Code (KSLOC)

### Intermediate COCOMO

Estimating Maintenance Effort for different development modes as follows:

Effort maintenance = A × EAF × (Size) B

Where,

A =Constants based on different classes of Software Projects

B=Economy or Diseconomy of Scale based on different classes of Software Projects

EAF= Effort Adjustment Factor

Size= No. of Source Lines of Code (KSLOC)

### Software LIfe-Cycle Management (SLIM)

Putnam approach is considered as an experimental software effort testing model. It is seen by Lawrence H. Putnam in 1978 as revolutionary work in a section of Processes regarding software Modelling. This approach of software estimation elaborates the effort and time that is supposed to be applied for a project of specified size. Software life-cycle management (SLIM) is closely related in software parametric to the COCOMO model.

## V Conclusion

Estimation process reflects the project progress reality. When the estimation is done accurately it produces the results accurately. It avoids the cost or budget as well as the schedule overruns. This process is quite simple which takes a few inputs. In this paper, this assessment framework helps the new inexperienced team improve project tracking and estimation skills. In future, this work can be carried on learning based methods which apply weights to the calculation of each software module based on priorities and criticalities.

## References

1. Bhandari, Sangeeta, "FCM Based Conceptual Framework for Software Effort Estimation", International Conference on Computing for Sustainable Global Development, 2016, pp 2585-2588.
2. Hill. R.Peter, "Practical Software Project Estimation –A Toolkit for Estimating Software Development Effort & Duration", Tata McGraw-Hill Edition, 2011, pp.75-79.
3. Kjetil Molokken et al., "A Review of Surveys on Software Effort Estimation", Proceedings of the International Symposium on Empirical Software Engineering (ISESE'03) 0-7695-2002-2/03.
4. Mendes et al., "A replicated assessment of the use of adaptation rules to improve Web cost estimation. In Empirical Software Engineering", Proceedings. 2003 International Symposium on ISESE 2003, ISBN: 0-7695-2002-2, 2003, pp. 100-109.
5. Mohammad D et al., "Management of Changes in Software Requirements during Development Phases", International Journal of Education and Management Engineering (IJEME), Vol.6, No.6, 2016, pp.12-26.
6. Mohammed Aljohani and Rizwan Qureshi, "Comparative Study Of Software Estimation Techniques", International Journal of Software Engineering & Applications (IJSEA), Vol.8, No.6, November 2017, pp 39-53.
7. Patil et al., "Generic Model of Software Cost Estimation: A Hybrid Approach". IEEE International Advance Computing Conference (IACC), 2014.
8. M.Pramod Kumar et al., "A Survey Report on Software Effort Estimation models and techniques", International Research Journal of Computer Science (IRJCS) ISSN: 2393-9842, Issue 12, Volume 3 (December 2016), pp 51-60.
9. Simon WU Iok Kuan, "FACTORS ON SOFTWARE EFFORT ESTIMATION", International Journal of Software Engineering & Applications (IJSEA), Vol.8, No.1, January 2017, pp 23-32.
10. P.K. Suri and Pallavi Ranjan, "Comparative Analysis of Software Effort Estimation Techniques", International Journal of Computer Applications (0975 – 8887) Volume 48– No.21, June 2012, pp 12-19.
11. Vahid et al., "Software Cost Estimation Methods: A Review", Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No. 1, 2011, pp.21-29.