

1986

# AN EMPIRICAL VALIDATION OF SOFTWARE COST ESTIMATION MODELS

Chris F. Kemerer  
*Carnegie-Mellon University*

Follow this and additional works at: <http://aisel.aisnet.org/icis1986>

---

## Recommended Citation

Kemerer, Chris F., "AN EMPIRICAL VALIDATION OF SOFTWARE COST ESTIMATION MODELS" (1986). *ICIS 1986 Proceedings*. 36.  
<http://aisel.aisnet.org/icis1986/36>

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 1986 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# AN EMPIRICAL VALIDATION OF SOFTWARE COST ESTIMATION MODELS

Chris F. Kemerer  
Graduate School of Industrial Administration  
Carnegie-Mellon University

## ABSTRACT

Practitioners have expressed concern over their inability to accurately estimate costs associated with software development. This concern has become even more pressing as these costs continue to increase. As a result, considerable research attention is now directed at gaining a better understanding of the software development process, as well as constructing and evaluating software cost estimating tools. This paper evaluates four of the most popular algorithmic models used to estimate software costs (SLIM, COCOMO, FUNCTION POINTS, and ESTIMACS).

Specifically, this paper addresses the following questions: 1) Are these models accurate outside their original environments and can they be easily calibrated? 2) Two of the models use source lines of code (SLOC) as an input, and two use inputs that are easier to estimate early in the project life cycle. Can the latter models be as accurate as the SLOC models, thus eliminating the need to attempt to estimate lines of code early in the project? 3) Two of the models are proprietary and two are not. Are the models that are in the open literature as accurate as the proprietary models, thus eliminating the need to purchase estimating software?

The methodology for evaluating these models was to gather data on completed software development projects and compare the actual costs with the *ex post* estimates obtained from the four models. Two tests were used to assess the accuracy of these models. The first was Conte's Magnitude of Relative Error (MRE) test, which divides the difference between the estimate and the actual effort by the actual effort, then takes the absolute value to eliminate problems with averaging positive and negative variances. The second test was to run simple regressions with the estimate as the independent variable and the actual effort as the dependent variable. The latter test was used for calibration and to judge the relative goodness of fit of the resulting linear models.

The source of the project data was a national computer consulting and services firm specializing in the design and development of data processing systems. The fifteen projects collected for this study covered a range of profit and not-for-profit applications. The majority of projects were written in COBOL, with an average size of approximately 200,000 source lines of code.

Analysis of the data yielded several practical results. First, models developed in different environments did not perform very well uncalibrated, as might be expected. Average error rates calculated using the MRE formula ranged from 85% to 772%, with many falling in the 500-600% range. Therefore, organizations that wish to use algorithmic estimating tools need to collect historical data on their projects in order to calibrate the models for local conditions. After allowing for calibration, the best of the models explain 88% of the behavior of the actual man-month effort in this data set.

The second estimation question concerned the relative efficacy of SLOC models versus non-SLOC models. In terms of the MRE results, the non-SLOC models (ESTIMACS and FUNCTION POINTS) did better, although this is likely due to their development in business data processing environments similar to that of the data source. In terms of the regression results, both COCOMO and SLIM had higher correlations than either ESTIMACS or FUNCTION POINTS. However, this conclusion must be made with reservation because the SLOC counts were obtained *ex post*, and are therefore likely to be much more accurate than SLOC counts obtained before a project begins.

The final research question on the relative accuracy of the proprietary and the non-proprietary models was not answered conclusively by this research. The proprietary SLIM model outperformed (in its regression coefficient of determination) the non-proprietary COCOMO model, while the non-proprietary FUNCTION POINTS model outperformed the proprietary ESTIMACS model for this data set.

This research has provided several important results regarding software metrics and models. First, Albrecht's model for estimating man-months of effort from the FUNCTION POINTS metric has been validated on an independent dataset. This is particularly significant in that FUNCTION POINTS have been proposed by IBM as a general productivity measure, and because prior to this there was only limited evidence for their utility from non-IBM sources. Second, algorithmic models, while an improvement over their basic inputs as predictors of effort, do not model the productivity process very well.

While improving estimation techniques within the industry is a worthwhile goal, the ultimate question must concern how the productivity of software developers could be improved. These questions are related in that the estimation models contain descriptions of what factors their developers believe affect productivity. The results of this study show that the models researched do not seem to capture the productivity factors very well. Further research needs to be done to isolate and measure these factors affecting systems professionals' productivity if the profession is to meet future challenges.

## Acknowledgement

The author wishes to thank Fred Forman, Charles Kriebel, and Mary Shaw for their assistance with this paper. Additional thanks to Douglas Putnam of Quantitative Software Management and Patrick Madden of Management and Computer Services for permission to use their models during this research.