Available online at www.sciencedirect.com
ScienceDirect
International Journal of Forecasting 23 (2007) 449–462
international journal of forecasting
www.elsevier.com/locate/ijforecast

Forecasting of software development work effort: Evidence on expert judgement and formal models

Magne Jørgensen *

Simula Research Laboratory, P.O. Box 134, NO-1325 Lysaker, Norway

Abstract

The review presented in this paper examines the evidence on the use of expert judgement, formal models, and a combination of these two approaches when estimating (forecasting) software development work effort. Sixteen relevant studies were identified and reviewed. The review found that the average accuracy of expert judgement-based effort estimates was higher than the average accuracy of the models in ten of the sixteen studies. Two indicators of higher accuracy of judgement-based effort estimates were estimation models not calibrated to the organization using the model, and important contextual information possessed by the experts not included in the formal estimation models. Four of the reviewed studies evaluated effort estimates based on a combination of expert judgement and models. The mean estimation accuracy of the combination-based methods was similar to the best of that of the other estimation methods.

Keywords: Judgemental forecasting; Combining forecasts; Comparative studies; Evaluating forecasts; Forecasting practice

1. Introduction

Clients require effort and cost estimates of software projects as inputs to investment analyses. Similarly, project managers require effort estimates to enable planning and to control the software development work. Unfortunately, many software development effort estimates are quite inaccurate. A recent review of estimation accuracy studies indicated that software projects expend on average 30–40% more effort than is estimated (Moløkken-Østvold & Jørgensen, 2003). There seems to have been no substantial improvement in estimation accuracy over the years. Software projects experience severe delivery and management problems due to plans based on overoptimistic effort estimates. The negative effects of overoptimism are accentuated by (i) software bidding rounds where those companies that provide overoptimistic effort estimates are more likely to be selected, and (ii) overconfidence in the accuracy of the estimates; for example, 90% confidence effort prediction intervals only include the actual effort 60–70% of the time (Jørgensen, Teigen, & Moløkken, 2004).

Software researchers have been addressing the problems of effort estimation for software development projects since at least the 1960s; see, e.g., Nelson

* Fax: +47 67 82 82 01.
E-mail address: magnej@simula.no.

0169-2070/$ - see front matter © 2007 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.ijforecast.2007.05.008

(1966). Most of the research has focused on the construction of formal software effort estimation models. The early models were typically regression-based. Soon, however, more sophisticated effort estimation models appeared, for example models founded on case-based reasoning, classification and regression trees, simulation, neural networks, Bayesian statistics, lexical analyses of requirement specifications, genetic programming, linear programming, economic production models, soft computing, fuzzy logic modeling, statistical bootstrapping, and combinations of one or more of these models. A recent review (Jørgensen & Shepperd, 2007) identified 184 journal papers that introduced and evaluated formal models for software development effort estimation. Many of these studies describe the re-examination and improvement of previously proposed estimation methods. Several estimation models have been included in commercially promoted tools. A survey by Moores and Edwards (1992) found that 61% of the IT managers in the UK had heard about at least one of these software development effort estimation tools. The use of formal estimation models has also been promoted by software process improvement frameworks and in software engineering education readings.

In spite of the extensive research into estimation models, the high degree of availability of commercial estimation tools that implement the models, the awareness of these estimation tools, and the promotion of model-based estimation in software engineering textbooks, software engineers typically use their expert judgement to estimate effort (Heemstra & Kusters, 1991; Hihn & Habib-Agahi, 1991).

The limited use of models may be a sign of the irrational behaviour of software professionals. It may, on the other hand, be the case that expert judgement is just as accurate or has other advantages that render the current low use of effort estimation models rational. This leads to the research questions of this paper: i) Should we expect more accurate effort estimates when applying expert judgement or models? ii) When should software development effort estimates be based on expert judgement, on models, or on a combination of expert judgement and models?

Extending Jørgensen (2004a), I review studies that compare the accuracy of software development effort estimates based on estimation models with those based on expert judgement and on a combination of these two approaches. The review process, limitations and results are included as Section 4. The factors examined in the review are derived from the discussion of the task of software development effort estimation in Section 2, and previous findings on the relative performance of model and judgement-based predictions are presented in Section 3. Section 5 provides concluding remarks about the implications of the findings of the review.

2. Software development effort estimation

For the purpose of this review, I separate expert judgement and model-based effort estimates based on the type of mental process applied in the "quantification step", i.e., the step where an understanding of the software development estimation problem is translated into a quantitative measure of the required effort. I define judgement-based effort estimates to be based on a tacit (intuition-based) quantification step, and model-based effort estimates to be based on a deliberate (mechanical) quantification step; see, for example, Hogarth (2001) for an elaboration of the meaning of these terms. The quantification step is the final step of the process, leading to an effort estimate for the total project or a project activity. If the final step is judgemental, the process is categorized as judgement-based. If the final step is mechanical, the process is categorized as model-based. There will be a range of quite different estimation processes belonging to each of the categories, i.e., neither expert judgement nor model-based effort estimation should be considered simply as "one method". When the outputs of two or more completed estimation processes are combined, we categorize the process as combination-based, and describe whether the combination step is judgemental or mechanical.

The term "expert" in this paper is used to denote all individuals with competence in estimating software development effort. In most studies, the expert is a software development professional, but we also use the term "expert" to denote, for example, a student with previous experience in effort estimation and the development of software for the type of task under consideration.

2.1. Expert judgement-based effort estimation processes

Most of the steps in the expert judgement-based effort estimation processes, e.g., the breaking down of the project into activities, may be explicit and can be