Industrial Use of Formal Software Cost Estimation Models: Expert Estimation in Disguise?

Magne Jørgensen, Simula Research Laboratory
Tanja Gruschke, Simula Research Laboratory, University of Oslo

***Abstract**: The goal of this paper is to propose and evaluate the hypothesis that software cost estimates based on formal estimation models are frequently expert estimation in disguise, i.e., that the cost estimates are not as mechanically derived as prescribed and assumed. We test implications of the hypothesis through discussion of related work and an empirical study of function point-based effort estimation of software projects. The actual effort estimates of the projects were compared with the effort estimates one would expect if the formal function point model was applied as prescribed. We observed several large deviations between the actual and the mechanically derived effort estimates, which we interpret as indications of a strong impact from expert judgment. Important limitations of our study are that the hypothesis is formulated vaguely, that there is not much evidence available, and, that we may have had a tendency to bias our search towards supporting evidence. More studies are therefore needed, preferably from independent researchers. If our hypothesis is correct, implementation of formal software cost estimation models should include means to avoid unwanted effects of initial beliefs and irrelevant information.*

1. Introduction

About 60% of all journal papers on software cost estimation focus on the introduction or evaluation of formal estimation models <*review paper in writing*>. Formal cost estimation models usually have as their goal to reduce the role of the human expert as provider of input to the models. An assumed benefit of the use of formal software cost estimation models is that it will generate estimates that are much less biased towards over-optimism and less affected by irrelevant information compared with expert judgment-based cost estimates. Examples of formal cost estimation models are parametric models, e.g., COCOMO (Boehm, Clark et al. 1996), function point-based models, e.g., IFPUG (Dekkers 1998), and analogy-based models, e.g., ANGEL (Shepperd and Schofield 1997).

This strong focus on the introduction and evaluation of formal cost estimation models may well be held to indicate that we, as researchers, have had a great interest in *how* the formal cost estimation models were used in the software industry, e.g., whether the models were used as intended or not and achieved the expected benefits. We were, therefore, upon reviewing journal software cost estimation papers, surprised to find no in-depth studies on this topic. In fact, we could not find *any* study that described *how* the models were applied in real project estimation contexts. The only evaluations of formal cost estimation models in real projects (and there are very few such studies) had a focus on which formal model that was applied, not how it was applied. In other words, our knowledge about how formal estimation models are applied in the software industry is very limited.

The first author of this paper has several years' experience as an advisor on software cost estimation in several companies, including the supervision of projects estimated by applying formal estimation models. As an advisor on such projects, he sometimes got the impression that the application of a formal cost estimation model masked an underlying expert judgment-based estimation process, i.e., the use of a formal estimation model was expert estimation in disguise. In particular, he frequently got the impression that the initial beliefs of the chief estimator or manager regarding costs had a strong impact on the actual cost estimated, regardless of the method of estimation used.

We were motivated by the experiences just described to examine the possibility that expert judgment has a strong influence on cost estimates that are, ostensibly, made solely as a result of the use of formal cost estimation models. In Section 2, we state a preliminary hypothesis about the effects of expert judgment in such cases of estimation and discuss related work that may support the hypothesis. In Section 3, we test one implication of the hypothesis on a small dataset collected in a Norwegian software company. Section 4 concludes.

2. A Preliminary Hypothesis

2.1 The Hypothesis

Our hypothesis is as follows:

Expert judgment frequently has a strong impact on the seemingly mechanically derived output of formal cost estimation models.

Note that we are principally concerned with the undesired effect of expert judgment that is not part and parcel of the estimation method. We are not concerned with the impact of the expert judgment that is required by some of the formal estimation models as input. One example of such an undesired effect is when an early belief, based on limited information about the project, results in the estimator adjusting the input values to ensure that the model output fits the initial belief.

The hypothesis is, in many ways, vague and weak. For example, (i) it uses the informal term "frequently", (ii) it does not specify what exactly is meant by "formal cost estimation model", and (iii) it does not specify the conditions under which the expert judgment is likely to have a strong effect, or *how large* the effect might be. It is, therefore, not obvious that it our hypothesis may be deemed truly scientific. Our reason for putting it forward, despite its stated failures with respect to strength and precision, is that, in the nature of the case, it is not possible to state initial hypotheses about human behavior with the precision that can be achieved with respect to hypotheses in the natural sciences. As a result we should allow, at least as a starting point for further refinements, a high degree of vagueness as long as the hypothesis is useful and it is possible to derive from it certain predictions that can be tested empirically, i.e. that are falsifiable.

A further way in which our hypothesis differs from hypotheses in natural science is that it knowledge of the hypothesis' content on the part of an actor may serve to undermine its validity to various degrees. Suppose that a person who is engaged in estimating the costs for a project becomes aware of our hypothesis. Such awareness may lead him to take action to reduce the risk of initial expert judgment having a strong impact on the outcome of formal model-based cost estimates. By contrast, hypotheses in natural science are unaffected by peoples' awareness of them.

We believe that our hypothesis is falsifiable, e.g., if several observational studies and experiments report that real-life software cost estimations are *not* strongly affected by initial expert judgment-based beliefs and much less biased towards over-optimism compared to expert judgment-based estimates. At the very least, such results would require substantial modification of the hypothesis.

2.2 Impact of Potential Expert Judgment

There are a number of ways in which seemingly mechanically derived cost estimates may be affected by expert judgment:

- Classification of elements used as input to the formal model, e.g., the classification of a module or transaction as being highly complex, may be influenced by the experts' initial beliefs about the required cost.
- Values used as input to estimation models may be adjusted so that the model output is more in line with the expert judgment output.
- Selection of historical projects relevant for the estimation may be affected by the estimator's initial belief about the size of the project to be estimated.

These potential effects, if they exist, do not have to be the result of conscious actions taken by the estimators, e.g., an estimator may believe that the estimate is a result of his/her use of the formal model, while in reality it is better described as an expert judgment-based estimate. Several studies, e.g., (Jørgensen and Sjøberg 2001) and (Northcraft and Neale 1987) , suggest that people are not aware of the all influencing factors and tend to rationalize how estimates were derived.

2.3 Implications of the Theory

If expert judgment has a strong impact on the output from formal cost estimation models we should, among other effects, observe certain testable implications:

Implication 1: There is little difference in the accuracy of estimates based on expert judgment and those based on the use of formal models.

Evidence 1: Earlier (Jørgensen 2004) we surveyed all empirical studies that compare expert judgment and formal model-based estimation. Of the 15 identified comparisons, only six are were based on real-life comparison of estimation accuracy. The results of many of these studies are difficult to interpret, due to methodological weaknesses; see the discussion in (Jørgensen 2004). However, as we interpret them, none of the studies provide a strong case for differences in estimation accuracy between expert judgment and formal model-based estimation. Most of the studies find no difference, and the studies that find differences find only small differences, e.g., that very large overruns may be somewhat reduced by applying formal cost estimation models. The observed lack of significant differences in, for example, average over-optimism between expert judgment and formal model-based estimates constitutes, we believe, support for our hypothesis.

An alternative interpretation of the results is that the estimates and the processes of expert judgment and formal cost models are different, and that the reason for similar estimation accuracy is simply that the quality of the two estimation methods is similar, or that factors other than the quality of the model used are responsible for the similarities in accuracy. This is possible, but if it were the case, we should then be able to identify underlying mechanisms that lead to similar quality of method. We find this difficult. In particular, we find it difficult to identify an underlying mechanism that would lead to such similarity in the observed levels of over-optimism. Formal cost estimation models should be neutral and not biased towards over-optimism.

Implication 2: There are biasing factors, irrelevant for the calculation of software cost estimates, which affect the output of both expert judgment and formal cost model-based estimation.

Evidence: We were unable to find any study on this topic within the software cost estimation domain. However, there is evidence for such factors in other domains. An example is the anchoring effect. In (Armstrong 2001) anchoring is described as "*the tendency of judges' estimates (or forecasts) to be influenced when they start with a 'convenient' estimate in making their forecasts. This initial estimate (or anchor) can be based on tradition, previous history or available data.*" In a previous paper, we documented that an irrelevant anchor value may, indeed, have a large, and unconscious, effect on software professionals' effort estimates (Jørgensen and Sjøberg 2004). When applying a formal estimation model, we would expect there to be no anchoring effect, i.e., we would expect that no formal estimation models use initial beliefs as input. Yet the opposite seems to be the case! For example, in (Northcraft and Neale 1987) a study is described where real-estate agents using formal calculation models to price a property were strongly affected by the anchoring effect. The real-estate agents did *not* notice this effect and believed that they were following their rather mechanical process of calculating the most likely market price. It is, of course, possible that there are differences between real-estate pricing and software cost estimation of which we are not aware. Nevertheless, this case serves to illustrate the contention that irrelevant factors may have an impact on seemingly mechanical estimation processes.

Is there any strong evidence against our hypothesis? It is possible, of course, to construct a situation in which estimators can be trained and instructed to follow an estimation process mechanically, without any element of expert judgment. Such situations seem, however, to be rare in the software industry and do not really falsify our hypothesis. What we hypothesize is that in industry settings, expert judgment frequently (not always!) affects estimations. Note that our failure to find evidence against the hypothesis may be a result of a strong tendency to search for supporting evidence and to reject negative information (Brehmer 1980). Further research, preferably by independent researchers, is required to assess the hypothesis with greater rigor.

3. An Empirical Study

If the hypothesis proposed in Section 2 is true we should observe that *actual effort MkII Function Point-based estimates frequently deviate a great deal from mechanically* *derived effort estimates*. Here, the mechanically derived effort estimate is interpreted as the output that results from the step-by-step estimation process described in Section 7 of the estimation text-book (Symons 1991).

To test this implication, we analyzed data from the estimation of five software projects of a large Norwegian software organization. The effort estimates of the projects were derived by applying the MkII function point method. Clearly, five software projects do not offer much data against which to test a hypothesis. The main contribution of the study may, therefore, be that it exemplifies how we believe that the hypothesis can be tested.

The company's process when applying the MkII FP deviates on minor points from those in (Symons 1991). The essence and the formalism of the estimation method, however, is the same. The organization's official estimation process was described by the estimation process owner as follows:

- Calculate the size of the system in MkII Function Points
    - Identify the logical transactions
    - For each logical transaction, count the number of input data element-types (Ni), entity-types referenced (Ne), and output data element-types (No).
    - Compute the unadjusted function points (UFP) as: 0.58*Ni + 1.66*Ne + 0.26*No.
- Compute the technical complexity adjustment (TCA) of the system based on answers on nineteen questions.
- Adjust for proportion of batch and on-line functionality
- Adjust for performance-driving risk factors and constraints
- Calculate the adjustment factor (AF) based on TCA, proportion of batch-functionality, and performance-driving factors/constraints.
- Calculate the adjusted function points as (AdjFP): UFP * AF
- Decide on the productivity (PROD) in MkII FP per work-hour based on the average productivity of projects applying the same development environment.
    - In the original method description <Symons> there are industry-standard productivity values to be used. The company, however, found these general values to be too inaccurate to be of any use.
    - Instead of the general industry values, the company used data collected from several previous projects of similar size and type. Based on these data, it had calculated its own standard productivity values, measured as AdjFP/Effort, dependent on development environment.
    - The recommended productivity values (the mean of previous projects) of the two most common development environments were:
        - PROD-Cobol: 0,15 AdjFP/work-hour
        - PROD-Powerbuilder: 0,35 AdjFP/work-hour
    - Within the interval of project sizes experienced in the organization, no significant difference in productivity related to size was observed, i.e., the productivity values were the same for all normal project sizes.
- Calculate the effort as AdjFP/PROD, where PROD depends on development environment

For each of the five projects we calculated the estimated effort by applying the MkII function point estimation process mechanically and then compared it with the effort estimate actually provided by the projects. See Table 1.

Table 1: Estimation Data (effort is measured in work-hours)

Projects	**Mechanically derived MkII FP effort estimates**	**Actual MKII FP effort estimates**	**Actual effort**
1	2112	993	1082
2	1392	1643	1696
3	1279	732	841
4	2720	1970	2531
5	5898	6240	6048

As can be seen in Table 1, there are strong deviations between most of the mechanically derived and actual effort estimates. This supports our hypothesis. Note that the fact that the actual estimates are closest to the actual effort does not necessarily mean that the mechanically derived estimates are less accurate. A project manager's task is, among other things, to manage the budget. Consequently, if the mechanically derived estimates had been applied as the actual effort estimates, they may have been more accurate (Jørgensen and Sjøberg 2001).

The main reason for the deviation between the mechanical and the actual use of the MkII FP process was the choice of the productivity factor (PROD), i.e., the last step of the estimation process. For example, in Project 3 the history-based productivity factor is 0,15 AdjFP/work-hour. The estimator, however, chose the productivity factor 0,26 AdjFP/work-hour. We can only speculate about the reasons for this large adjustment, but it is possible that the recommended productivity factor led to estimates that deviated from the estimators "expert judgment" and that the productivity factor was therefore adjusted. Alternatively, the estimators may have believed that the standard productivity was not applicable to their projects and may have chosen the productivity factor based on expert judgment. In both cases, the actual estimation process is strongly dominated by expert judgment.

There are strong limitations of this study. A better test of our hypothesis would be based on much more information about how the formal cost estimation process was used, about the initial beliefs of the total cost estimators and how historical data was applied.

3. Conclusion

This paper puts forward a preliminary hypothesis to the effect that formal software cost estimates are frequently expert estimation in disguise, even when the estimators state that they have followed a formal estimation process. We support this preliminary hypothesis with evidence from other studies and results from one empirical study. The present formulation of the hypothesis is vague and evidence in its favor is currently weak. Hence, further research is required.

Is it necessarily a bad thing that seemingly mechanical estimation processes are strongly affected by expert judgment? Perhaps not, as long as mechanical estimation processes perform as poorly as they seem to. However, problems may arise when formal cost estimation models are believed (falsely) to lead automatically to unbiased and history-based estimated. This belief may have the consequence that means to reduce the bias from, and effect of, irrelevant information such as customer expectation and initial beliefs based on limited information, are not considered important when formal cost estimation models are applied.

References:

Armstrong, J. S., Ed. (2001). Principles of forecasting: A handbook for researchers and practitioners. International Series in Operations Research & Management Science. Boston, Kluwer Academic Publishers.

Boehm, B., B. Clark, et al. (1996). "The COCOMO 2.0 software cost estimation model: A status report." American Programmer **9**(7): 2-17.

Brehmer, B. (1980). "In one word: Not from experience." Acta Psychologica **45**: 223-241.

Dekkers, C. A. (1998). "Function points: refresher course." Managing System Development **18**(7): 8-10.

Jørgensen, M. (2004). "A review of studies on expert estimation of software development effort." Journal of Systems and Software **70**(1-2): 37-60.

Jørgensen, M. and D. I. K. Sjøberg (2001). "Impact of effort estimates on software project work." Information and Software Technology **43**(15): 939-948.

Jørgensen, M. and D. I. K. Sjøberg (2004). "The impact of customer expectation on software development effort estimates." International Journal of Project Management **22**: 317-325.

Northcraft, G. B. and M. A. Neale (1987). "Amateurs, experts, and real estate: An anchoring-and-adjustment perspective on property pricing decision." Organizational behavior and human decision processes **39**: 84-97.

Shepperd, M. and C. Schofield (1997). "Estimating software project effort using analogies." IEEE Transactions on Software Engineering **23**(11): 736-743.

Symons, C. R. (1991). Software sizing and estimating - Mk II FPA. Chichester, U.K., John Wiley&Sons.