Empirical Software Engineering, 9, 297–314, 2004.

Regression Models of Software Development Effort Estimation Accuracy and Bias

MAGNE JØRGENSEN magne.jorgensen@simula.no
Simula Research Laboratory

Editors: Marian Petre, David Budgen and Jean Scholtz

Abstract. This paper describes models whose purpose is to explain the accuracy and bias variation of an organization's estimates of software development effort through regression analysis. We collected information about variables that we believed would affect the accuracy or bias of estimates of the performance of tasks completed by the organization. In total, information about 49 software development tasks was collected. We found that the following conditions led to inaccuracies in estimates: (1) Estimates were provided by a person in the role of "software developer" instead of "project leader", (2) The project had as its highest priority time-to-delivery instead of quality or cost, and (3) The estimator did not participate in the completion of the task. The following conditions led to an increased bias towards under-estimation: (1) Estimates were provided by a person with the role of "project leader" instead of "software developer". (2) The estimator assessed the accuracy of own estimates of similar, previously completed tasks to be low (more than 20% error). Although all variables included in the models were significant ($p < 0.1$), the explanatory and predictive power of both models was poor, that is, most of the variance in the accuracy and bias of estimates was not explained or predicted by our models. In addition, there were several important threats to the validity of the coefficients suggested by the models. An analysis of the estimators' own descriptions of the reasons for achieved estimation accuracy on each task suggests that it will be difficult to include all important estimation accuracy and bias factors in regression-based models. It is, for this reason, not realistic to expect such models to replace human judgment in estimation uncertainty assessments and as input to plans for the improvement of estimates. It is, nevertheless, possible that the type of formal analysis and regression-based models presented in this paper may, in some cases, be useful as support for human judgment.

1. Introduction

Software organizations should know how to develop software and how much effort they need to develop it. However, it is also important for them to be aware of how uncertain their effort estimates are, so that they can, for example, analyze the risk of exceeding the project budget or decide whether there is a need to take action to reduce the uncertainty in use of effort.

Possible approaches to assessing the uncertainty of an effort estimate are, for example:

- *Approach 1:* Application of effort prediction intervals[1] based on uncertainty properties of formal estimation models of most likely effort. This approach includes the use of prediction intervals of regression models and prediction intervals based on bootstrapping of analogy-based models, as described and evaluated in Angelis and Stamelos (2000).

- *Approach 2:* Application of effort prediction intervals based on previous accuracy of similar estimation tasks. This approach includes the use of empirical and parametric distribution of previous estimation accuracy, as described and evaluated in Jørgensen and Sjøberg (2003).

- *Approach 3:* Application of the output from a regression analysis-based model of estimation accuracy. Notice that while the uncertainty assessment models based on Approach 1 include only variables important for prediction of the most likely effort, regression models based on Approach 3 include variables important for prediction of the most likely estimation accuracy, that is, Approach 3 may be described as a more direct approach to predicting estimation accuracy.

- *Approach 4:* Effort prediction intervals based on human judgment. Studies of the properties of effort prediction intervals for software development that are based on human judgment can be found in Connolly and Dean (1997) and Jørgensen et al. (2003).

To the best of our knowledge, the most common approach is Approach 4, that is, assessments of uncertainty based on human judgment. The use of Approach 4 is frequently informal, for example the estimator judges based on non-explicit and non-recoverable processes that he/she is "almost sure" that the actual effort of a task will be between 8000 and 12,000 work-hours. A major problem with effort prediction intervals based on human judgment is that they are typically much too narrow to reflect the stated confidence level (Jørgensen et al., 2004). For example, when the estimator claims to be 90% certain or "almost certain" of including the actual effort in the effort prediction, the actual frequency of including the actual effort is typically 50–60% (Jørgensen et al., 2004). Variations of the elicitation processes of prediction intervals based on human judgment have been shown to remove the bias towards over-confidence in experimental conditions with immediate and precise estimation accuracy feedback (Jørgensen and Teigen, 2002). Unfortunately, preliminary results from an on-going real-life evaluation (conducted by the author of this paper) indicate that the benefits from the proposed changes in the uncertainty elicitation process may be lower in cases with lack of, or late, feedback on estimation accuracy. This means that important changes in how feedback on estimation accuracy typically is provided may be needed to improve uncertainty assessments that are based on human judgment.

One advantage of effort prediction intervals based on human judgment seems to be that they use the available information about uncertainty more efficiently than do model-based effort prediction intervals (Jørgensen et al., 2004), for example software professionals are able to make use of uncertainty information specific to a particular project or development environment, while models are, of necessity, based on general relationships. Model-based effort prediction intervals, on the other hand, seem prone to providing meaningless wide effort prediction intervals when data sets are small and important information about uncertainty is not covered by the included variables. However, effort prediction intervals that are based on models do have an important advantage: they are unbiased, being unsusceptible to the over-confidence to which human estimators can fall prey. In short, present knowledge on uncertainty assessment approaches suggests that we may have to choose between over-optimistic uncertainty assessments based on human judgment, and inefficient assessments based on models (Jørgensen et al., 2004).

This paper evaluates the use of Approach 3. This approach has, to the best of our knowledge, not been evaluated and may, therefore, have properties different from other approaches to model-based uncertainty assessments. In particular, we hoped that the inclusion of variables relevant to uncertainty, as opposed to variables related to most likely effort (as in Approach 1), would improve the regression model.

2. The Study

2.1. Previous Work

The first step in building a regression model, when applying Approach 3, was to identify the variables that have the potential to predict the accuracy of estimates. There have been several empirical studies on the differences in accuracy between different estimation models with different variables and parameters, for example (Briand et al., 1999) and (Jørgensen, 1995), and the difference in estimation accuracy between formal estimation models and expert judgment, see Jørgensen (2004) for an overview. However, there seem to be few software development studies that identify variables that affect accuracy and bias other than estimation model and process. Standish Group (www.standishgroup.com) reports that the main source of improvement in the accuracy of estimates from 1994 to 1998 was the shift towards smaller projects. The results reported in Gray et al. (1999) suggest that over-estimation was connected with changes in small modules and the development of screens, while under-estimation was connected with changes in large modules and the development of reports. Earlier, we reported from a study at Ericsson Design Center in Norway (Jørgensen and Løvstad, 2002) that estimation accuracy was affected by the project priority. Projects that placed a high priority on time-to-delivery were more likely to be under-estimated than those that focused on costs.

2.2. Data Collection Process and Measures

We studied a medium-sized Norwegian web-development company. Over a period of approximately 10 months we collected information about tasks with an estimated effort of more than one workday and duration of less than approximately four calendar months. In total, information about 49 tasks was collected. The Chief Project Manager of the company was in charge of the data collection. He asked the developers to complete one questionnaire before starting the task and another one after the task was completed. We have a low number of observations ($n = 49$) in relation to the variation of data values. To increase the possibility of significant inclusion of the categorical variables, therefore, we joined the values of the nominal-scaled variables with the closest mean estimation accuracy values until all nominal-scaled variables were binary variables with the values 0 and 1. For example, originally, the variable "Customer Priority" had three categories: time, cost and quality. The accuracy values of the tasks with priority on cost and quality had the closest mean values and were, for this reason, combined in the category cost/quality. The variable "Customer Priority", as applied in this analysis, has consequently the two priority categories cost/quality and time.

The information collected before a task started, including examples of the motivation for the inclusion of the variable, was as follows:

- Company role of the estimator (Project Leader = 0, Developer = 1). A project leader might make better estimates, due to having had more practice at making them and having received better feedback. Typically, the software developers are only responsible for estimating own work.

- Brief description of the task (maximum 10 lines).

- The estimators' assessment of the complexity of the development task (Low = 0, Medium/High = 1). The interpretation of "complexity of a task" was left to the developer. A complex task may be more complex to estimate.

- Type of contract (Payment per hour = 0, Fixed price = 1). When an estimate is input to a fixed price contract, more emphasis may be placed on quality estimation work.

- The estimators' assessment of how important the task is for the customer (Low/Medium = 0, High = 1). A high priority task may be subject to more thorough estimation work and, for this reason, achieve higher estimation accuracy.

- The priority that the customer assigns to the project (Cost or Quality = 0, Time-of-delivery = 1). As described in the introduction, the fact that time-of-delivery is of paramount importance for a project seems to be a predictor of poor estimation accuracy.

- The estimators' level of knowledge about how to perform the task (Low/Medium = 0, High = 1). Knowing a great deal about how to perform a task may improve the accuracy of the estimate.

- The estimators' planned participation in the completion of the task (Participates = 0, Does not participate = 1). Estimating one's own work might be easier than estimating other people's work.

- The estimators' perception of the typical accuracy with which he has made estimates concerning similar tasks (0–20% = 0, More than 20% = 1). The historical accuracy of estimates for similar tasks might be an indicator of future accuracy.
- The estimated effort in work hours. As described in the introduction, there are studies suggesting that small tasks are, on average, over-estimated and large tasks under-estimated.

After the task was completed the estimators provided:

- The actual effort in work hours.
- Comments on the actual use of effort (free text).
- Descriptions of unexpected problems during the execution of the task (free text).
- Reasons for high or low estimation accuracy (free text).

We apply the following accuracy measures in our model building:

- Magnitude of Relative Error (MRE) = abs[(Actual Effort − Estimated Effort)/Actual Effort].
- Relative Error (RE) = (Actual Effort − Estimated Effort)/Actual Effort; A positive RE-value corresponds to an under-estimate, and a negative value to an over-estimate.

While MRE is a measure of the absolute estimation accuracy, RE shows the direction of the estimation deviation and can, consequently, be applied to model the estimation bias. The MRE measure may not always be optimal for describing the estimation accuracy. For example, MRE penalizes over-estimation more heavily than under-estimation. No matter how much the actual effort is underestimated MRE cannot exceed one, whereas overestimation leads to MRE values with no upper limits. We evaluated, for this reason, the use of the alternative accuracy measures:

1. Balanced Relative Error = |Act − Est|/min(Act, Est), see discussion in Miyazaki et al. (1994) and Jørgensen and Sjøberg (2003).
2. Logarithmic Relative Error = |ln(Act/Est)|.

These accuracy measures may have properties that lead to better explanatory models for some data sets. In our case, however, they did not lead to any improvement and we therefore kept the more common MRE-measure.

3. Results

3.1. Descriptive Statistics

Table 1. Value distributions.

Variable	Values
Company role	31% of the tasks were estimated by project managers, 69% by software developers.
Task complexity	29% of the tasks were categorized as having low complexity, 71% as having medium or high complexity.
Contract type	45% were fixed price tasks, 55% were paid at an hourly rate.
Importance	26% of the tasks were assessed to have a low or medium customer importance, 74% a high importance.
Customer priority	67% of the tasks placed a high priority on cost (given an acceptable level of quality) or quality, and 33% on time-of-delivery (given an acceptable level of quality).
Level of knowledge	37% of the tasks were estimated by estimators with a self-assessed low or medium level of knowledge (knew little or something) about how to solve the task, 63% with a high level of knowledge (knew a great deal about how to perform the task).
Participation	59% of the tasks were estimated by estimators who participated in completing the work and 41% by estimators who did not participate.
Previous accuracy	73% of the tasks were estimated by estimators who believed that their typical estimation accuracy on similar tasks was "low" (less than 20%) and 27% "high" (more than 20%).
Estimated effort	The mean estimated effort was 89 work hours, with a minimum of 6 work hours and a maximum of 700 work hours. The relation between estimated effort and estimation accuracy is not likely to be linear, for example it is not likely that increasing the estimated effort from 1000 to 1100 work hours would have the same impact on estimation accuracy as an increase from 10 to 110 work hours. For this reason, we log-transformed the estimated effort when building the regression model that is, we used log(Estimated Effort) as a variable.
Actual effort	The mean actual effort was 117 work hours, with a minimum of 14 work hours and a maximum of 998 work hours.
Estimation accuracy	The mean absolute relative estimation error (MRE) was 27%. The mean relative estimation error (RE) was 10%, that is, there was a bias towards under-estimation.

3.2. Regression Models

We applied linear regression models to model the accuracy and bias of the estimation. There are, of course, many other types of model and our choice is motivated principally by the need for simple models and analysis tools to support this preliminary attempt at understanding estimation accuracy relationships.

The regression models of MRE and RE were developed by applying stepwise regression with backward elimination. We used an alpha-value of 0.1 to remove variables. The resulting regression models, and connected significance values, from this process are:

$$\begin{array}{llll} \text{MRE} = 0.14 + & 0.13 \text{ Company Role} + & 0.13 \text{ Participation} + & 0.13 \text{ Customer Priority} \\ (p = 0.03) & (p = 0.08) & (p = 0.07) & (p = 0.09) \\ \text{RE} = 0.12 - & 0.29 \text{ Company Role} + & 0.27 \text{ Previous Accuracy} & \\ (p = 0.05) & (p = 0.004) & (p = 0.01) & \end{array}$$

The MRE-model suggests that the estimation error increases in the following circumstances: when a task is estimated by a person with the company role of software developer rather than project leader, when the estimator does not participate in the completion of the task rather than estimating his own work, and when the customer prioritizes time-to-delivery rather than quality or cost. These relationships accord, to a large extent, with what we expected based on earlier experience and common sense.

The different signs of the variables in the RE-model make the model more difficult to interpret. A "mechanical" interpretation of the models is the following:

- If the estimator is a project leader (Company Role = 0) and the previous estimation error was less than 20% (Previous Accuracy = 0), then there is an average under-estimation of 12%.

- If the estimator is a project leader (Company Role = 0) and the previous estimation error was more than 20% (Previous Accuracy = 1), then there is an average under-estimation of 39%.

- If the estimator is a software developer (Company Role = 1) and the previous estimation error was less than 20% (Previous Accuracy = 0), then there is an average over-estimation of 17%.

- If the estimator is a software developer (Company Role = 1) and the previous estimation error was more than 20% (Previous Accuracy = 1) then there is an average under-estimation of 10%.

This suggests, somewhat surprisingly, that although project leaders seem to have had more accurate estimates than software developers (according to the MRE-model), they may also have been more prone to under-estimation. The situation most likely to lead to under-estimation seems to be when the estimator is a project leader and the previous estimation accuracy on similar tasks is poor. We discussed these results with the project leaders of the company. They explained the under-estimation by the project leaders as a consequence of their role, because they are responsible for efficient project work. If project estimates are a little bit on the low side, the project members will be pushed to work more efficiently. If this explanation is correct, it points to difficulties in interpreting the term "estimate", that is, what the project leaders referred to as an estimate was in reality a mix of "most likely" and "cost reducing planned effort".

Note that additional variables would probably have been included in our models if we had had more observations, that is, our regression-based analyses do not exclude the importance of the non-included variables. The low number of MRE-observations per variable value combination means also that some of the regression coefficient may be misleading. In short, the models should be interpreted with great care. We discuss threats to validity further in the following section.

3.3. Validity of the Models

There are several threats to the validity of the regression models, for example over-fitting of the model, violation of the regression assumptions, and lack of predictive validity. This section discusses the validity of the models and the included variables.

3.3.1. Over-Fitting

Tables 2 and 3 show an analysis of the best sub-set of variables for each number of variables (Vars) included in the model. The square of the multiple correlation coefficient (R^2) indicates the fit of the model. When there are few observations and many variables, as in our study, it is important to apply the R^2-adjusted instead of R^2.R^2-adjusted indicates the proportional reduction in the mean square deviation between actual and estimated effort, rather than the reduction in the sum of the squares.

As can be seen in Table 2, the model with the highest R^2-adjusted value is the one we presented earlier, with the variables Company Role, Customer Priority and Participation. Inclusion of more variables increases the unadjusted, but not the adjusted R^2, that is, more variables would lead to a high risk over over-fitting. Table 2 also indicates the level of robustness of the variables included in the model. The variables included in our MRE-model are included in all best subset-models with number of variables greater than three, that is, the variable inclusion seems to be robust.

The adjusted R^2 is low (11.1%), even for the best model. This means that the majority of the variance in estimation accuracy is not described by our model. A low

Table 2. Best subset regression with response MRE.

Vars	R^2	R^2-adjusted	Comp. role	Task compl.	Contr. type	Import.	Cust. prior.	Level of knowl.	Partic.	Prev. acc.	ln(Estim)
1	9.0	7.1					X				
2	11.4	7.6			X		X				
3	16.6	11.1	X				X		X		
4	18.3	10.8	X		X		X		X		
5	20.0	10.7	X		X	X	X		X		
6	20.5	9.1	X		X	X	X		X		X
7	21.4	7.9	X	X	X	X	X		X		X
8	21.4	5.7	X	X	X	X	X	X	X		X
9	21.4	3.2	X	X	X	X	X	X	X	X	X

Table 3. Best subset regression with response RE.

Vars	R^2	R^2-adjusted	Comp. role	Task compl.	Contr. type	Import.	Cust. prior.	Level of knowl.	Partic.	Prev. acc.	ln(Estim)
1	12.2	10.3	X								
2	24.3	21.1	X							X	
3	26.5	21.6	X						X	X	
4	27.7	21.2	X	X					X	X	
5	28.7	20.4	X	X				X	X	X	
6	28.8	18.7	X	X	X			X	X	X	
7	28.9	16.7	X	X	X			X	X	X	X
8	28.9	14.7	X	X	X	X		X	X	X	X
9	28.9	12.5	X	X	X	X	X	X	X	X	X

adjusted R^2 does, however, not entail that the model we found lacks validity in explaining the relationship between the mean estimation accuracy and the included variables.

Table 3 shows that we did not select the model with the highest adjusted R^2, that is, inclusion of the variable Participation would improve the adjusted R^2 from 21.1 to 21.6. However, the difference in explanatory power is not large and we decided, for reasons of simplicity, to keep the original model. Similar to the MRE-model, the inclusion of the variables seems to be robust, that is, the two variables included in our RE-model are part of all best subset-models with number of variables greater than two. The adjusted R^2 of the selected model is low, although higher than that of the MRE-model. Most of the estimation bias variance is therefore not explained by the RE-model.

3.3.2. Regression Assumptions

Regression models are based on the assumption that the errors, that is, the residuals, are normally distributed, independent, with a constant variance and mean value that equals zero. A validation of the regression models should therefore include an examination of the residuals. Figures 1 and 2 show plots useful for such examinations based on the output from the statistical tool MINITAB.

Figure 1 suggests that the regression assumptions are to some extent violated. The normal plot and the histogram suggest that the residuals are close to, but not completely, normally distributed. In particular, there is a bias towards negative residuals, that is, towards too high MRE-values provided by the model. The "I Chart" (the individual observation chart) shows that the mean residual value is close to zero and that there are no large outliers, for example no observations outside the three sigma limit indicated by the values UCL and LCL. The chart depicting residuals vs. fits suggests that the variance of the residuals increases with increasing MRE-value, that is, the variance is not constant. This suggests that the model is, to some extent, misspecified. We tested different types of variable transformations, for

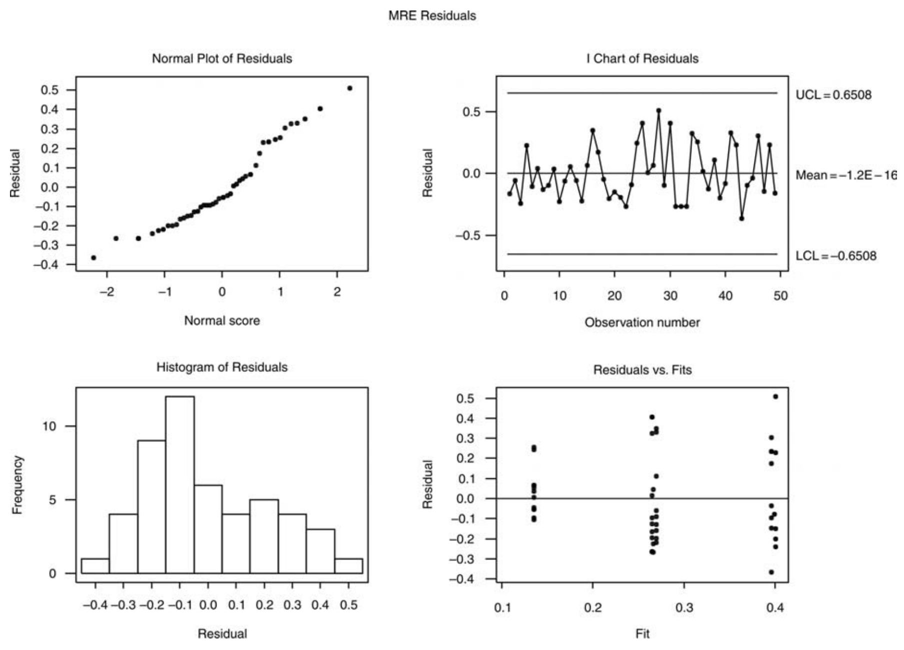

Figure 1. Residuals of the MRE-model.

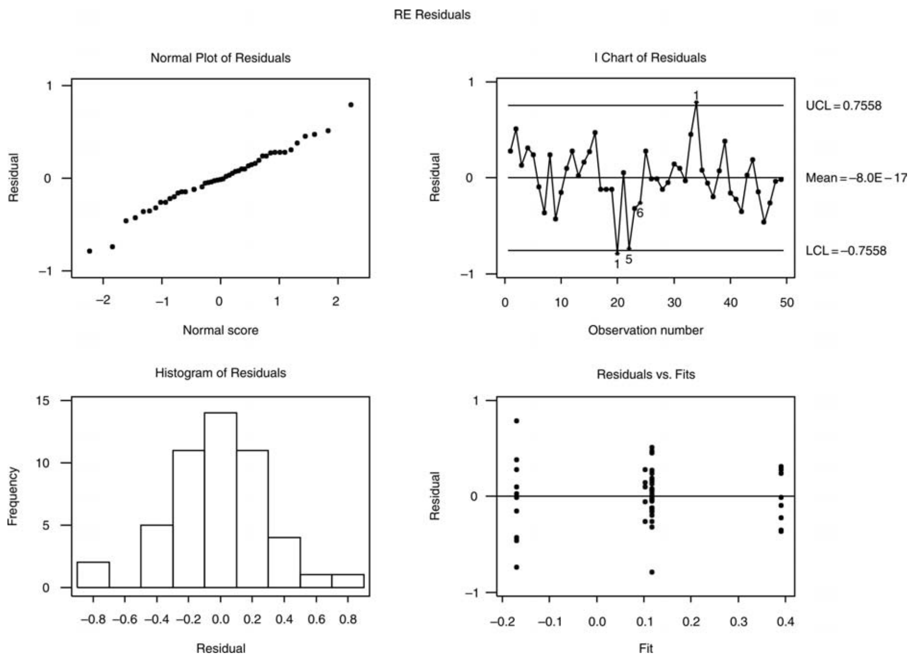

Figure 2. Residuals of the RE-model.

example logarithmic, to reduce this problem of correlation between residuals and "Fit". Unfortunately, similar violations of regression conditions were present for the transformations we tested.

Figure 2 suggests that the residuals have a distribution similar to a normal distribution. The "I Chart" shows that there are three outliers, that is, observations with values close to the three sigma limit. The chart depicting residual vs. fits suggests that the variance of the residuals increases with low RE-values, that is, the variance is not constant. Similar to the MRE-model residual analysis, there is a need for further investigation of model improvements and variable transformations.

3.3.3. Understanding the Residuals

Tables 4 and 5 display important information about estimation accuracy (excerpts from the estimators' own descriptions of reasons for high/low estimation accuracy) for the tasks with the 10 highest MRE and RE-model residuals. Appendix 1 gives/states the reasons for high/low estimation accuracy for all tasks, including those described in Tables 4 and 5.

Table 4. Ten highest MRE-model residuals.

Task Id	MRE-residual	Estimated MRE	Actual MRE	Estimated effort	Actual effort	Reasons for estimation accuracy/inaccuracy
48	− 0.54	0.13	0.67	15	45	Problems with sub-contractor.
2	− 0.54	0.13	0.67	30	18	The software developer worked faster than expected.
44	− 0.46	0.13	0.59	6	15	Unexpected technical and data formatting problems.
49	− 0.44	0.26	0.70	300	998	Poor quality of input to estimation process. Lack of proper error correction and test processes.
1	− 0.39	0.52	0.91	67	35	The customer changed the scope of the task during its execution, that is, less functionality than originally planned.
18	0.39	0.39	0.0	420	422	Experience from previous completion of similar tasks for the same customer.
46	− 0.37	0.26	0.63	10	27	Installation problems caused by insufficient software licenses.
47	− 0.37	0.26	0.63	160	437	Forgot to include important activities in the estimate, for example support on user acceptance test.
8	0.36	0.52	0.16	174	151	High flexibility in how to implement the required functionality.
13	0.35	0.39	0.04	50	48	The same update of software is completed three times every year.

Table 5. Ten highest RE-model residuals.

Task Id	RE-residual	Estimated RE	Actual RE	Estimated effort	Actual effort	Reasons for estimation accuracy/inaccuracy
45	− 0.79	− 0.17	0.62	70	186	There were no new tasks available when the task was finished. Therefore, the developer spent additional time on improving the quality and added a few non-specified features.
2	0.79	0.12	− 0.67	30	18	The software developer worked faster than expected.
1	0.74	− 0.17	− 0.91	67	35	The customer changed the scope of the task during its execution, that is, less functionality than originally planned.
47	− 0.51	0.12	0.63	160	437	Forgot to include important activities in the estimate, for example support on user acceptance test.
44	− 0.47	0.12	0.59	6	15	Unexpected technical and data formatting problems.
3	0.46	− 0.17	− 0.63	700	431	Much more than expected of the previously developed software could be reused.
43	− 0.45	0.12	0.57	20	47	Frequent changes in the requirement specification.
33	− 0.38	− 0.17	0.21	220	279	Forgot to include installation and user training in the estimate.
19	0.36	0.39	0.03	200	206	⟨reasons not provided⟩

Tables 4 and 5, together with the data in Appendix 1, suggest that most of the reasons provided by the estimator for high/low estimation accuracy are different from, or only vaguely related to, the variables in our MRE- and RE-models. In fact, it is hard to see how a formal model could include all important estimation accuracy and bias variables without an unrealistically high number of observations. For example, it may be difficult to model the effect on estimation accuracy of “there were no new tasks available when the task was finished” at the time of estimation without knowledge of the totality of present and future tasks at the company. Alternatively, there is a need for adjustment of the actual effort to reflect the differences between the estimated and completed task.

It is interesting to note the diversity of the reasons for low/high estimation accuracy described in Appendix 1. There seems, however, to be two frequent reasons for estimation performance: (1) A change of requirement specification is a major reason for inaccurate estimates, and (2) similarity to previously completed tasks is a major reason for accurate estimates. The inclusion of variables that represent these two reasons may lead to an improvement in our models of estimation accuracy and bias. Another interesting observation from Appendix 1 is that low estimation accuracy may frequently be the result of project control issues, for example lack of requirement management processes, not necessarily poor estimation processes. In other words, an accurate estimate requires not only good estimation skills, but good cost management skills, as well.

3.3.4. Predictive Value of the Models

To assess the predictive value of the model we performed a linear, cross-validation-based, discriminant analysis of how the selected variables of our regression models contributed to the prediction of high and low MRE and RE. We defined high (low) MRE to be more (less) than 30% deviation from the actual effort and high (low) RE to be more (less) than 10% under-estimation. The values used for division of the variable values into categories, that is, 30 and 10%, were approximations of the mean MRE- and RE-values. The selected variables should have a predictive value at least better than a random model to support our model variable selection, that is, the prediction models should predict more than 50% of the estimation categories correctly. Note that this is not a completely proper cross-validation procedure, because the selection of the variable and the choice of MRE- and RE-categories are based on the full data set. However, the "leaving-one-out procedure" implemented in the cross-validation we performed would probably not lead to important changes, that is, in practice there would probably be no important difference in results.

Tables 6 and 7 suggest that the predictive value was only slightly better than a random model. The MRE-category model had 63% correct predictions based on the selected MRE-regression model variables and the RE-category model had 59% correct predictions based on the selected RE-regression model variables. These

Table 6. Cross-validated, discriminant analysis of MRE-category based on the variables: Company role, participation, and customer priority.

Predicted MRE-category	True MRE-category	
	Low MRE ($<30\%$ deviation)	High MRE ($\geq 30\%$ deviation)
Low MRE	18	6
High MRE	12	13

Table 7. Cross-validated, discriminant analysis of RE-category based on the variables: Company role and previous accuracy.

Predicted RE-category	True RE-category	
	Low RE ($<10\%$ under-estimation)	High RE ($\geq 10\%$ under-estimation)
Low RE	20	16
High RE	4	9

results correspond with the regression model results. The selected variables seem to be a valid part of the models, but they are not capable of predicting more than a small proportion of the variance in estimation accuracy and bias.

Perhaps the most striking type of incorrect prediction is that of "Low RE" when the true RE-category is "High RE", that is, the discriminant analysis indicates that the factors predicting high under-estimation are the least understood factors in our models and, consequently, need extra attention.

4. Conclusion

Based on our analysis of 49 software tasks within one organization we found that estimation error increased when the task was estimated by a software developer (vs. project leader), when the estimator did not participate in the completion of the task (vs. estimation of own work), and when the customer prioritized time-to-delivery (vs. quality or cost). In addition, we found that project leaders were more prone to under-estimation than software developers, and that under-estimation was more likely when the estimator believed that his/her estimates of similar tasks had been inaccurate ($> 20\%$ deviation).

This type of information may be useful as a starting point for organizations when improving their estimation processes and assessing the uncertainty of estimates of future software development tasks. However, it is essential that the models be interpreted with great care. For example, the use of project leaders to estimate tasks was connected with an increased accuracy of the estimates. To apply this information properly, however, a better understanding of the underlying reason for this finding may be required. Possible reasons are, amongst others, that there are differences between tasks estimated by the project leaders and the other tasks, that project leaders are more motivated for high estimation accuracy, that project leaders provides estimates more frequently and receives more estimation feedback, that project leaders are less prone to forget/under-estimate non-technical activities, or that the same personal skill that led them to the role of project leader is itself a good predictor of high estimation accuracy. More data and better analyses are needed to provide better cause–effect analyses of estimation accuracy variables.

Most of the variance in estimation accuracy was not explained by our regression models. We obtained results similar to those gained from evaluating the other model-based approaches (Approaches 1 and 2). The analysis of the model residuals and the estimators' own descriptions of reasons for low/high estimation accuracy suggest that we cannot expect formal models to explain most of the estimation accuracy and bias variation, unless the amount of observations and variables is unrealistically high. For example, many important reasons for low estimation accuracy are connected to seldom-occurring events and cost management issues. In our opinion, formal estimation accuracy models should therefore be applied as input to uncertainty assessments based on human judgment and actions for improving the estimation process, not as replacements for human judgment.

As a result of the disappointing results from the evaluation of the formal effort uncertainty assessment models, in the future we will work on how to combine the advantages of the formal models with the advantages of approaches based on human judgment. In other words, we intend to focus on how to avoid the tendency of software professionals to be over-confident, without losing the flexibility and efficiency of uncertainty assessments based on human judgment.

Appendix 1

The table is sorted by RE-values. For more information about the tasks, please request the author for the full data set.

Task id	Estimated effort	Actual effort	MRE	RE	Reasons for high/low estimation accuracy
1	67	35	0.91	– 0.91	The customer changed his opinion regarding design and functionality.
2	30	18	0.67	– 0.67	The developer worked much faster than expected.
3	700	430.5	0.63	– 0.63	Much more than expected of the previously developed software could be reused.
4	37.5	23.5	0.60	– 0.60	The test of the program was combined with other tests. This reduced the effort spent on test preparation.
5	22.5	17	0.32	– 0.32	The customer provided all necessary information. Easy task.
6	30	25	0.20	– 0.20	Change of self-developed software.
7	26	22	0.18	– 0.18	Before estimating the task, "test-development" of some of the functionality was conducted.
8	174	150.5	0.16	– 0.16	Good knowledge and flexibility concerning how to implement the requirements.
9	16	14	0.14	– 0.14	The estimate was the most likely effort + 25% additional effort.
10	25	22	0.14	– 0.14	Much time spent on the analysis phase, risk budget added, and good knowledge of how to solve the task.
11	20	18.5	0.08	– 0.08	Much experience with similar tasks.
12	15	14	0.07	– 0.07	The accuracy was good in spite of change requests, because a risk budget was added.
13	50	48	0.04	– 0.04	Knowledge of the task was very good. The same update is conducted three times a year.
14	18	17.5	0.03	– 0.03	Task similar to other tasks recently completed.
15	25	25	0.00	0.00	<no reason provided>
16	40	40	0.00	0.00	<no reason provided>
17	30	30	0.00	0.00	<no reason provided>
18	420	422	0.00	0.00	Similar task completed, for the same customer, recently.
19	200	206	0.03	0.03	<no reason provided>
20	80	83	0.04	0.04	Very similar task for another customer recently completed.
21	48	50.5	0.05	0.05	Everything went as planned.
22	25	27	0.07	0.07	Easy task.

Task id	Estimated effort	Actual effort	MRE	RE	Reasons for high/low estimation accuracy
23	24	26	0.08	0.08	Easy to understand the implications of the task.
24	50	55	0.09	0.09	Flexible task. Used the estimated effort and then stopped.
25	18	20	0.10	0.10	Long experience with this type of task.
26	50	56	0.11	0.11	More testing and meeting with the customer than expected.
27	15.5	18	0.14	0.14	Problems with software performance.
28	60	72	0.17	0.17	Too low a fixed price, not reflecting the risks, was contracted with the customer.
29	15	18	0.17	0.17	The installation took more time because an important feature had not been developed.
30	144	178	0.19	0.19	No important problems, but many small changes in the requirement specification.
31	60	75	0.20	0.20	Experience from estimating similar projects. More effort than expected on understanding a feature of the software development tool.
32	40	50	0.20	0.20	Additional effort was added to the most likely effort because of high risk and vague specification.
33	220	279	0.21	0.21	Estimate did not include requirement changes and training.
34	370	494	0.25	0.25	Error in requirement specification assumptions (assumed that a feature that had to be developed was already part of the application).
35	157	210	0.25	0.25	Poor quality of data and data models.
36	10.5	14.5	0.28	0.28	Task larger than expected.
37	50	71.5	0.30	0.30	Changes in requirement specification.
38	10	14.5	0.31	0.31	More time to understand the existing system than expected.
39	120	186.5	0.36	0.36	Lower quality than expected from other software applications.
40	50	80	0.38	0.38	Problems with sub-contractor.
41	80	130	0.38	0.38	Good relationship was important to this important customer. A too low estimate, not reflecting the risk of the task, was developed.
42	123	200	0.39	0.39	More discussion than expected with the customer. Increase in functionality.
43	20	47	0.57	0.57	Frequent change requests from the customer.
44	6	14.5	0.59	0.59	Unexpected technical and data formatting problems.
45	70	186	0.62	0.62	There were no new tasks available to start on when this was finished. Therefore, time was spent on improving the quality beyond the minimum and adding a few non-specified features.
46	10	27	0.63	0.63	Problems with installation due to software license changes.
47	160	437	0.63	0.63	Forgot to estimate effort on assisting the customer in acceptance testing.
48	15	45	0.67	0.67	Problems with sub-contractor.
49	300	998	0.70	0.70	Poor quality of input to estimation process. Lack of proper error correction and test processes.

Note

1. An effort prediction interval is an interval with boundaries "minimum effort" and "maximum effort" and a corresponding confidence level. For example, a project manager may estimate that the most likely effort required to complete a project is 10,000 work hours, and add that he/she is 90% confident that the effort required will turn out to be between 8000 (minimum effort) and 12,000 (maximum effort) work-hours. Here, the interval [8000, 12,000] work hours is the 90% confidence prediction interval of the effort estimate.

References

Angelis, L., and Stamelos, I. 2000. A simulation tool for efficient analogy based cost estimation. *Empirical Software Engineering* 5: 35–68.

Briand, L. C., El Emam, K., Surmann, D., Wieczorek, I., and Maxwell, K. D. 1999. An assessment and comparison of common software cost estimation modeling techniques. *International Conference on Software Engineering*, Los Angeles, USA, ACM, New York, pp. 313–323.

Connolly, T., and Dean, D. 1997. Decomposed versus holistic estimates of effort required for software writing tasks. *Management Science* 43(7): 1029–1045.

Gray, A., MacDonnell, S., and Shepperd, M. 1999. Factors systematically associated with errors in subjective estimates of software development effort: The stability of expert judgment. *Sixth International Software Metrics Symposium, IEEE Comput. Soc.* Los Alamitos, CA, USA, pp. 216–227.

Jørgensen, M. 1995. Experience with the accuracy of software maintenance task effort prediction models. *IEEE Transactions on Software Engineering* 21: 674–681.

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., Løvstad, N., and Moen, L. 2002. Combining quantitative software development cost estimation precision data with qualitative data from project experience reports at Ericsson Design Center in Norway. *Empirical Assessments of Software Engineering (EASE)*, Keele, UK.

Jørgensen, M., and Sjøberg, D. I. K. 2003. An effort prediction interval approach based on the empirical distribution of previous estimation accuracy. *Journal of Information Software and Technology* 45: 123–136.

Jørgensen, M., and Teigen, K. H. 2002. Uncertainty intervals versus interval uncertainty: An alternative method for eliciting effort prediction intervals in software development projects. *International Conference on Project Management (ProMAC)*, Singapore, pp. 343–352.

Jørgensen, M., Teigen, K. H., and Moløkken-Østvold, K. J, 2004. Better sure than safe? Overconfidence in judgment based software development effort prediction intervals. *Journal of Systems and Software* 70(1–2): 79–93.

Miyazaki, Y., Terakado, K., Ozaki, K., and Nozaki, H. 1994. Robust regression for developing software estimation models. *Journal of Systems and Software* 27: 3–16.

Magne Jørgensen received the Diplom Ingeneur degree in Wirtschaftswissenschaften from the University of Karlsruhe, Germany, in 1988 and the Dr. Scient. degree in informatics from the University of Oslo, Norway in 1994. He has 10 years industry experience as consultant and manager. He is now professor in software engineering at University of Oslo and member of the software engineering research group of Simula Research Laboratory in Oslo, Norway. His research interests include software cost estimation, uncertainty assessments in software projects, expert judgment processes, and, learning from experience.