A Preliminary Study of Sequence Effects in Judgment-based Software Development Work-Effort Estimation

Stein Grimstad, Magne Jørgensen
Simula Research Laboratory, P.O. Box 134, NO-1325 Lysaker, Norway
{steingr,magnje}@simula.no

ABSTRACT

Context: Software development effort estimates are often inaccurate, and this inaccuracy cause problems for the clients as well as the providers. Consequently, we need more knowledge about the estimation processes, so that we can improve them.
Objective: This study investigates how initial judgment-based estimation of work effort in software development affects subsequent, unrelated estimation work.
Method: Fifty-six software professionals from the same company were allocated randomly to two groups. One group estimated the most likely effort required to complete a small software development task, while the other group estimated the effort required to complete a large task. After that, all the subjects estimated the effort required to complete the same medium-sized task. We replicated the experiment in another company (with 17 software professionals).
Results: We found that sequence effects may have a strong impact on judgment-based effort estimates. Both in the first experiment and in the replication, the subsequent estimates were assimilated towards the subjects' initial estimate, i.e., the group that began with a small task supplied, on average, lower estimates of the medium-sized task than the group that began with the large task.
Conclusion: Our findings suggest that knowledge about sequence effects may be important in order to improve estimation processes. However, currently we have a quite incomplete understanding of how, when and how much sequence effects affect effort estimation. Consequently, further research is needed.

Keywords: software effort estimation, judgment-based estimation, sequence effects.

1. BACKGROUND

Several research studies have found that accurate software estimation is an important factor for success in software development projects; see e.g. (Lederer and Prasad, 1995; Ropponen and Lyytinen, 1997). Unfortunately, a recent survey (Moløkken and Jørgensen, 2003) reports that the average estimation error is about 30% in software development projects. We may conclude that there is an urgent need for more accurate software estimates. A better understanding of the processes of human judgment that are relevant to software effort estimation may be important in order to reduce estimation error, because human judgment plays a central role in almost all software estimation. The relevance for judgment-based estimation processes (e.g. expert estimation) is obvious. Not so obvious, but important nevertheless is its relevance for formal estimation models; it typically plays an important role in providing input to the models, selection of estimation model, etc.

Human judgment has been studied extensively in other fields of research, such as cognitive and social psychology, experimental economics, forecasting, jury decisions, and consumer research. These studies have revealed numerous shortcomings in human judgment; see e.g. (Koehler and Harvey, 2004; Tversky and Kahneman, 1974). Previous studies have demonstrated that several of these issues are relevant to software estimation. For example, estimates are usually over-optimistic (Bergeron and St-Arnaud, 1992), over-confident (Jørgensen et al., 2004), inconsistent (Grimstad and Jørgensen, 2007), assimilated towards judgmental anchors (Aranda and Easterbrook, 2005) and affected by irrelevant information (Jørgensen and Grimstad, 2008). Knowledge about such shortcomings increases our understanding of the estimation process, and is important input for the development and improvement of estimation methods. For example, in (Jørgensen and Grimstad, 2008) we show that estimators who have a vested interest in the outcome of the estimation process are typically poor at making realistic estimates. It would, therefore, be wise to avoid using such persons as estimators.

In this study we focus on whether software professionals' current effort estimation work may be affected by unrelated estimation work that they have recently conducted. Will, for example, their estimates be too optimistic if they have recently estimated a very small task? Research on human judgment suggests that this may be the case. There is substantial evidence that activating a construct in one task, often referred to as contextual priming, increases the likelihood that it will later affect a subsequent, unrelated task; see e.g. (Higgins, 1996). For example, (Thomas et al., 2007) found that the duration of a just-completed anagram task affected the prediction of the duration of the next, structural different, anagram task, and that this led to over-optimistic estimates when the previous duration was shorter and to overly pessimistic estimates when it was longer than the current task.

In software estimation, it is common for software development tasks to be estimated directly after each other. Typically, a project is broken down into subtasks, which are then estimated in separate estimation sessions.

If the order in which the tasks are estimated affects the estimates, as research in other fields suggests, there may be orders in which tasks are estimated that are likely to provide more realistic estimates than others. However, few estimation methods address the order in which the tasks are estimated. This is, perhaps, not surprising, because we are not aware of any research studies that have investigated the effects of the sequence in which tasks are estimated in the context of software engineering. However, it would be useful to conduct such studies. In addition to offering practical advice, they may also contribute to a better understanding of the underlying steps involved in the cognitive processes of software estimation. The lack of previous research and the practical and scientific relevance of the topic motivated the research question in this study:

RQ: How does the sequence in which software development tasks are estimated affect the estimates in the judgment-based estimation of the most-likely software development effort?

We conducted a quasi-experiment to investigate our research question. The experiment was designed to test how estimating a large task vs. estimating a small task affects the subsequent estimation of a medium-sized task. We analyzed how the estimators' competence level impacted the sequence effect, and we replicated the experiment in order to test the robustness of the results on different subjects.

The remainder of this paper is organized as follows. In Section 2 we present the experiment, the replication and the results. In Section 3, we discuss the limitations of the study, suggest guidelines, and, discuss possible explanations for the effect. Section 4 summarizes.

2. EXPERIMENTS

2.1 Experimental Design

Subjects

The experiment was conducted as a part of an in-company estimation seminar in a medium-sized consultancy company located in eastern Norway. The company's main focus is web-based development for external clients. The 56 subjects described themselves as developers, designers, architects, technology experts, project leaders, and managers, i.e. as experienced software professionals who had different backgrounds and fields of expertise. However, most of the subjects had a technical education and most had previously been involved in the estimation of several software development projects. The subjects did not receive any payment. Instead, we used the results to illustrate key issues in the seminar.

Material

We created three independent requirement specifications. Each described a software development task. The amount of functionality and complexity in each requirement specification differed. We characterized the tasks as small (TS), medium (TM) and large (TL), according to the amount of effort required to complete the tasks. The tasks were based on the use of standard web-related technologies and there were no constraints regarding development tools and methodology. This was to ensure that most subjects had sufficient competence for meaningful estimation work. Two of the tasks (TM and TL) were based on real-world software specifications, while the remaining task (TS) was created for experimental purposes. The specifications were written in natural language. See Table 1 for an overview of the tasks.

Table 1 Estimation Tasks

Task Id	Task Size	Description
TS	Small	A simple web system for the registration of seminar participants. Participants register on the web by submitting their email address and a registration code. The system confirms that the data is registered. There are no data validation (duplicate check, etc). The data is stored in a database. Generation of reports, such as attendee lists, is done manually, i.e. by querying the database.
TM	Medium	A web-based library system that contains information about scientific articles. Users and administrators can view an information page about each scientific article that is registered in the system, search for articles, see a printer-friendly display of the search results and the information pages, register new scientific articles (some data validation is done during registration), and perform simple user management (administrators can register, edit and remove other administrators).
TL	Large	A web-based system that manages experiments and other studies. Users can view an information page about each study. The page contains information about the study design, the results, involved persons, related research articles, etc. Users can perform advanced searches, sort the search results, see the results in a printer-friendly display, generate graphical reports, etc. Administrator users can upload and manage files, add/delete/edit studies, and perform simple user management. The system requires some integration with other systems.

Procedure
The subjects were randomized into two groups (Group TS-TM and TL-TM) by their physical location in the seminar room (every second subject was allocated to the same group). The subjects received a booklet that contained requirement specifications. The subjects were instructed to estimate the development tasks in the booklet in the same order as they appeared, and they were not allowed to go back and change previous, already completed, estimates. We collected the booklets when the allocated time had expired.

Each group was asked to estimate two of the three requirement specifications; see Table 2. One group initially estimated the large task, while the other group initially estimated the small task. Subsequently, both groups estimated the middle-sized task. The tasks were estimated by expert judgment, and the subjects did not have access to any additional information. The subjects did not implement the tasks. We performed a pilot study prior to the experiment, and we had previously used variants of the requirement specifications in experiments. We used our experience from the pilot and the previous experiments to design this experiment, e.g. when allocating time to complete the estimation tasks.

Table 2 Treatment

Estimation Task	Group TS-TM	Group TL-TM
Estimation task 1	TS	TL
Estimation task 2	TM	TM

When the subjects had completed the estimation work, we asked them to assess their competence level related to estimation of the software development tasks. The competence categories were described as follows (translated):

"My competence to estimate this work is: Very good – Good – Acceptable – Weak".

Replication
We replicated the experiment in another in-company estimation seminar. The subjects were 17 experienced software professionals (mainly developers) from a software department in a large company that is located in the middle of Norway. The company's main focus is in-house development and maintenance work for their company.

We attempted to replicate all relevant aspects of procedure from the first experiment, including the tasks, the allocation of the tasks to treatment, and the amount of time that was allocated.

Results
The results of the first experiment are displayed in Table 3 and Figure 1, and those of the replication in Table 4 and Figure 2.

The inter-estimator agreement is low in both the experiment and the replication. This a common finding in estimation studies; see e.g. (Grimstad and Jørgensen, 2007; Kusters et al., 1990). There are several possible reasons for this, some of which are related to internal inconsistency (Grimstad and Jørgensen, 2007), and variations in productivity (Brooks, 1975). Neither is it surprising that there appear to be systematic inter-company differences. It is likely that there are certain company-specific issues that can affect both the effort used and the estimation, related, for example, to clients, personnel skills, and the development process.

We did not exclude potential outliers. Instead, we based the analysis on the median values (Kruskall-Wallis tests) in order to increase the robustness. The effect was stronger when we based the analysis on mean values.

Table 3 Experiment: Median Most Likely Estimates (work-hours)

Group	N	Estimate of TS	Estimate of TL	Estimate of TM
TS-TM	28	24,0	N/A	95,0
TL-TM	28	N/A	550,0	195,0

Table 4 Replication: Median Most Likely Estimates (work-hours)

Group	N	Estimate of TS	Estimate of TL	Estimate of TM
TS-TM	28	20,0	N/A	72,0
TL-TM	28	N/A	230,0	90,0

Figure 1 Experiment: Median Most Likely Estimates of Medium Task vs. Group

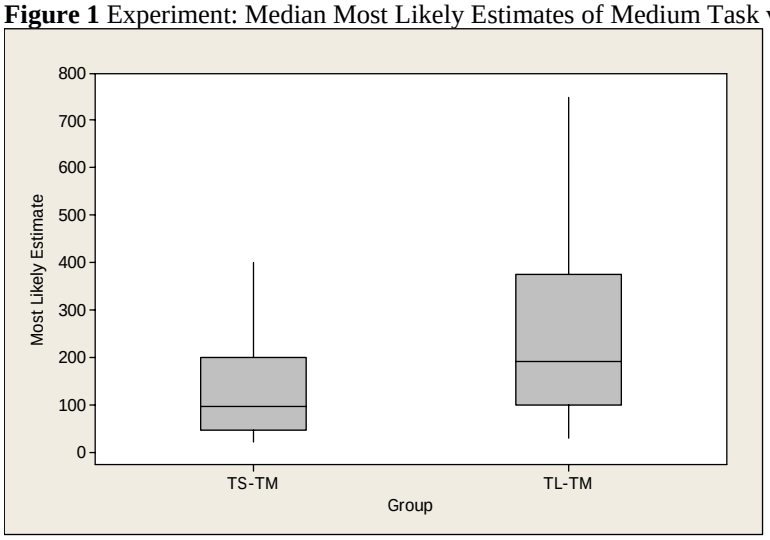
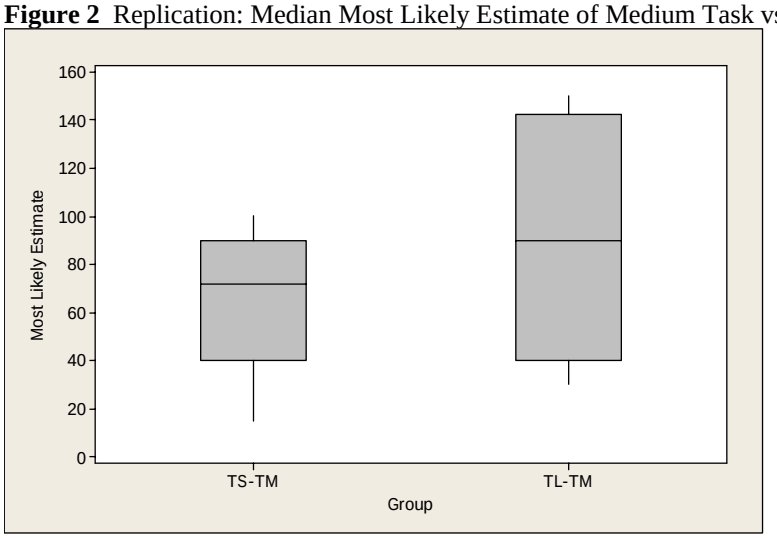


Figure 2 Replication: Median Most Likely Estimate of Medium Task vs. Group



The results show that the subjects that initially estimated the small task (Group TS-TM) submitted, on average, lower effort estimates for the medium task than the subjects that initially estimated the large task (Group TL-TM) (median estimates of the middle-sized task of 95,0 vs. 190,0 work-hours in the experiment, and 72,0 vs. 90,0 work-hours in the replication). Statistical analysis shows that the effect of task order on the estimates is statistically significant (p=0.01). The effect is not statistically significant (p=0.3) in the replication. Still, we believe that the replication strengthens the results from the original experiment, because the results clearly point in the same direction. It would be worthwhile to repeat the replication with a higher number of subjects, to determine whether the results are significant, because the replication only used 17 subjects, whereas the original experiment used 56

In both cases, the relative effect size of the treatments is medium large according to the classification in (Cohen, 1992)[1] (Cohen's d is 0,68 in the experiment and 0,60 in the replication).

Level of Competence

Table 5 report the median estimates for the different groups and competence categories for the first experiment, and those of the replication are reported in Table 6. We have combined categories A, B and C to form the category "High skill", and compared it to category D ("Low skill"). Other combinations, e.g. comparing category A and B to category C and D gave similar results. Note that the groups are based on self-assessed competence. Consequently, these results should only be seen as an indication of the relationship between competence level and sequence effects. One of the subjects in the first experiment did not report competence level and was therefore excluded from the analysis.

The results show that the level of competence did not reduce the sequence effect in the first experiment. In the replication, only the less skilled developers seemed to be affected by the sequence. However, the results from the replication should be implemented with great care as the number of subjects in each group is low. We consequently believe that the correct interpretation of the results related to competence level is that selecting the

[1] We have based the previous statistical analysis on the median values. Therefore, we have not removed potential outliers from the dataset. However, these outliers might impact Cohen's d as this is a measure that is based on the mean values. The effect sizes should therefore be interpreted with some care.

most competent software estimators is not a safe way to remove sequence effects. The results point in the same direction as the findings in studies on the impact of irrelevant information on expert judgment based estimation of software development work. For example, in (Jørgensen and Grimstad, 2008) we found that the level of competence reduced the impact on the estimates of irrelevant information related to variation in wording, but it did not remove it.

Table 5 Experiment: Median Most Likely Estimates (work-hours) and Skill Level

Group	Skill	N	Estimate of TS	Estimate of TL	Estimate of TM
TS-TM	High	10	22,0	N/A	60,0
TS-TM	Low	18	36,0	N/A	135,0
TL-TM	High	11	N/A	350,0	125,0
TL-TM	Low	16	N/A	650,0	225,0

Table 6 Replication: Median Most Likely Estimates (work-hours) and Skill Level

Group	Skill	N	Estimate of TS	Estimate of TL	Estimate of TM
TS-TM	High	6	13,0	N/A	80,0
TS-TM	Low	3	24,0	N/A	40,0
TL-TM	High	4	N/A	145,0	40,0
TL-TM	Low	4	N/A	300,0	125,0

3. DISCUSSION

It is well-known that human judgment can be unreliable. The results demonstrate that judgment-based software effort estimation is no exception, and suggest that sequence effects can have a large impact on effort estimates of software development tasks. The results also illustrate that it is difficult to predict the impact of sequence effects, i.e. we are not able to explain satisfactorily the effect size in the original experiment and the replication.

However, the results should be interpreted with care because of the limitations to this study. They include issues related to the following:

- Time pressure. The subjects had about 20 minutes to estimate the two tasks. This restricted the amount of in-depth analysis that was possible. It may be that a thorough and time-consuming analysis of the necessary development work reduces sequence effects. There are, for example, studies that have found that primacy effects and judgmental anchoring increase in magnitude when there is increased time pressure under certain conditions; see e.g. (Kruglanski and Freund, 1983). In our experience, the time that was allocated to the estimation work in the experiment is typical for this type of estimation work in field-settings. Nevertheless, there are clearly real-world estimation situations in which more time is spent on the estimation work.
- Estimation method. The subjects estimated the tasks by expert judgment. They were not allowed to use other estimation methods, such as group estimation and methods based on formal estimation models, and they were not allowed to discuss the estimation work with colleagues. Estimation methods may diverge with respect to the type and magnitude of sequence effects. For example, the justification component in discussion-based processes may moderate the effect of sequence effects. Consequently, we may have studied one of the estimation methods that is most likely to be affected by sequence effects.
- Laboratory context. The experiment was conducted as a part of an estimation seminar, so the subjects were not functioning in their usual work context. As a result, they did not have access to historical estimation data, or any other information, apart from what they could remember. It may be the case that estimation in field situations are less affected by sequence effects than estimation in laboratory studies. We have, for example, found in an unpublished study that the effect of judgmental anchors may be significantly lower in some field situations than that which is typically found in laboratory studies.
- Estimation tasks. There were no variations in estimation tasks in the experiment. Studies have shown that the sequence effects can lead to assimilation and contrast; see e.g. (Stapel and Koomen, 1998), and that there are large variations in effect sizes. Consequently, it is not unlikely that other estimation tasks, e.g. tasks that are less similar, will give completely different results.
- Estimation accuracy. We can only speculate on how sequence effects would have affected the estimation accuracy if the participants had implemented the tasks that they estimated. It is intuitive to think that starting by estimating the largest task would improve the estimation accuracy, because the average effort

estimates of the subsequent estimation work increased and it is well-known that effort estimates are often too optimistic. However, there are many factors that potentially affect estimation error, including the estimate itself, and it is difficult to accurately predict how estimation error will be impacted by a specific factor (Grimstad and Jørgensen, 2006).

In order to address some of the limitations related to time pressure, estimation method, laboratory context and estimation accuracy, we analysed the data from a previous study. In a field experiment (Jørgensen, 2004), seven estimation teams from a large company estimated two real-life software development projects. Each estimation team applied a top-down estimation strategy on one project and a bottom-up estimation strategy on the other. The estimators were allowed to telephone people in their own company (e.g., other software developers who had relevant experience), and to collect documents from their own offices or computers. In addition, they had access to the company's online database of completed projects. The projects that they estimated had already been completed by other teams in the company, but the participants in the experiment did not know anything about the projects. The actual effort used for the first project was 1340 work-hours, and the actual effort used for the second project was 766 work-hours.

Unfortunately, all the estimation teams estimated the projects in the same order. Obviously, this complicates the analysis of sequence effects. However, we based the analysis on the finding that estimates are likely to be assimilated towards previous estimates (see Section 2). A possible consequence of this is that estimates are likely to be too pessimistic when the previous estimate is larger than the current one, as is the case in the study reported herein. We therefore expected that the estimates of the second project would be less over-optimistic than the estimates of the first project. The results support this hypothesis. The median estimates of the first project are on average 14% too optimistic, and the median estimates of the second estimate are 15% too pessimistic. However, there are numerous limitations to this analysis, and the results should be implemented with great care.

In order to examine issues related to estimation tasks characteristics and sequence effects, we re-analysed data from an experiment that was conducted in order to investigate the level of inconsistency in expert judgment-based estimation processes (Grimstad and Jørgensen, 2007). In the experiment, seven experienced software professionals were selected based on their estimation accuracy in a previous study. Each subject estimated 60 software development tasks. There were three estimation sessions with one month between each session. The subjects completed 20 estimation tasks in each session. Six tasks were estimated twice (with at least one month in-between), e.g. the fourth estimation task in the first session is identical to the sixth estimation task in the third session. The estimation tasks that preceded the tasks that was estimated twice differed, e.g. the third estimation task in the first session were not the same task as the fifth task in the third session. Consequently, there are 42 pairs (seven subjects * six tasks) of corresponding estimates (data points) where we can test the impact of the initial estimates on subsequent estimation work.

The experiment was based on the effort estimation of development tasks on an existing web-based database system written in Java. The subjects had previously been involved in the development of the system, i.e. they were familiar with the domain and the technologies. The tasks were estimated by expert judgment. The subjects had access to the system documentation, but not to the source code. The tasks have not yet been implemented. The development tasks varied in perceived size and complexity, i.e. the estimates of work-effort necessary to complete the development tasks varied from 0,5 to 100 work-hours.

Table 7 reports how the estimation that preceded the two estimates of the same development task impacted the subsequent estimation work. The first column reports the magnitude of the size difference of the initial estimates. The second column reports how many data points that is included in the categories defined by the first column. The third column reports the magnitude of sequence effects that we found among the data points reported in column two, i.e. whether the first of the two estimates of the same development task, by the same subject, is higher (lower) than the second estimate if the estimate that preceded the first estimate is higher (lower) than the estimate that preceded the second estimate. The last column report how much the initial estimates affected the subsequent estimation work, i.e. the median relative difference of the two estimates of the same task (by the same subject). Three of the data points were excluded because the initial estimates were identical.

It is not surprising that the results show that the probability of observing sequence effects increase when the difference between the initial estimates increases. However, it is perhaps somewhat more surprising that the median effect size seems to be about 40% in each group. This finding should not be emphasized too strongly, because there are relatively few data points and the variation in effect size is large (from 5% to more then 60%) Consequently, we believe that the results suggest that we may be good to identify estimation situations where it is likely that sequence effects will occur, but we are not able to predict how much the estimates will be impacted.

Table 7 Initial Task Size and Sequence Effects

Size difference of the initial estimates	N	Data points with sequence effects	Median effect size of the sequence effect
Small (less than 4 work-hours)	21	9 (43%)	41%
Medium (4 – 8 work-hours)	10	6 (60%)	36%
Large (more than 8 work-hours)	8	7 (88%)	40%

Sequence effects may be hard to avoid in real-world estimation situations. It is, for example, quite common that software development projects are re-estimated during the project execution. This typically means that all the uncompleted development tasks that are included in the project are estimated within a short timeframe. Our results suggest that it is likely that such estimates will be affected by sequence effects. Unfortunately, it is difficult to predict the impact of the sequence effects, e.g. related to effect size. At present, our best advice is as follows:

In situations in which it is unlikely that the estimates will be over-optimistic, it may be best to start by estimating medium-sized and medium-complex tasks. However, when it is reason to suspect that estimates will be too optimistic, it may be best to start by estimating the largest and most complex tasks.

However, a better understanding of the sequence effects might allow us to go beyond these simple guidelines and offer advice on how to neutralize the effect. This will require knowledge about how, when, why, how much and under what conditions sequence effects affects judgmental estimation processes.

Most of the numerous models and theories that explain aspects of human judgment and decision making, such as the social judgment model proposed by Mussweiler in (Mussweiler, 2003), assumes that almost all human judgment is based on comparisons. An essential step in comparative judgment processes is to find a relevant reference that with which the current judgment task can be compared. The selection of reference for comparison will often impact the outcome of the judgment process, see e.g. (Herr, 1986; Jacowitz and Kahneman, 1995). A possible explanation of our results is consequently that the initial estimation task was used as a reference in the estimation of the subsequent task. However, there are many cognitive mechanisms that can cause the reference for comparison to produce the sequence effects that we observed in our experiment. It may, for example, be the case that selection of a large reference for comparison increased the focus on complexity related attributes, such as quality and testing, in the judgment-based estimation processes. Selection of a small reference might have increased the focus on attributes such as simplicity and rapid development. Unfortunately, our study does not allow us to discriminate between the different cognitive mechanisms.

We believe that the main contribution of this paper is to demonstrate that sequence effects may have a large impact on software estimation. However, our current understanding of the phenomena is quite incomplete. Carefully designed studies are needed to reveal the mechanics that are involved and how they interact. Clearly, further research is needed.

4. SUMMARY

The typical approach to the estimation of work effort in software development is based on the decomposition of projects into subtasks. These subtasks are usually estimated in a rather arbitrary sequence. However, research in other fields suggests that the sequence may be important. For example, studies on forecasting have found that initial predictions can strongly affect subsequent, even unrelated, predictions. We designed an experiment to test whether such sequence effects occur in a typical software effort estimation situation.

In a laboratory-based experiment, we divided 56 software professionals randomly into two groups. One group started by estimating a small, and the other a larger, software development task. Subsequently, all the software professionals were asked to estimate the work effort of the same medium-sized task. We found that the estimates of the medium-sized tasks were assimilated towards the initial estimates, i.e., the group that initially estimated a small task submitted, on average, lower estimates of the medium-sized task than the group that initially estimated a larger task. Selecting the most competent developers as estimators is not a safe way to remove the effect. We replicated the experiment and obtained similar results.

There are several limitations to the experiment. For example, the experiment was conducted in a laboratory setting, there were time pressures that prevented in-depth analyses, and there was a lack of variation in the tasks that were estimated. It is not unlikely that other estimation contexts would have yielded different results. For example, the estimation method that all the software professionals used in the experiment was that of expert judgment. It may be the case that other estimation methods are more (or less) robust with respect to sequence effects.

Despite these limitations, our study indicates that sequence effects are more important than they are currently treated as being in software effort estimation research and practice. Such sequence effects may affect whether estimates are too optimistic, too pessimistic or realistic, and a better understanding of the sequence effects may help us to understand and improve software professionals' estimation performance. Currently, our understanding of how, when, and how much, sequence effects affect effort estimation is poor, and. further research is needed.

At present, our best advice is that software professionals should start with effort estimates of medium-complex, medium-sized sub-tasks of the project, or, with large and complex tasks if there is a tendency towards over-optimistic estimates.

References

Aranda, J. and S. Easterbrook, 2005. Anchoring and adjustment in software estimation, Software Engineering Notes 30(5): 346-355.

Bergeron, F. and J. Y. St-Arnaud, 1992. Estimation of information systems development efforts: a pilot study, Information and Management 22(4): 239-254.

Brooks, F. P. (1975). The mythical man-month: Essays on Software Engineering. Reading, Mass., Addison-Wesley.

Cohen, J., 1992. A Power Primer, Psychological Bulletin 112: 155-159.

Grimstad, S. and M. Jørgensen, 2006. A framework for the analysis of software cost estimation accuracy. ACM/IEEE International Symposium on Empirical Software Engineering, Rio de Janeiro, Brazil, ACM New York, NY, USA.

Grimstad, S. and M. Jørgensen, 2007. The Impact of Irrelevant Information on Estimates of Software Development Effort. The Australian Software Engineering Conference, Melbourne, Australia, IEEE Computer Society.

Grimstad, S. and M. Jørgensen, 2007. Inconsistency of expert judgment-based estimates of software development effort, Journal of Systems and Software 80(11): 1770-1777.

Herr, P. M., 1986. Consequences of priming: Judgment and behaviour. , Journal of Personality and Social Psychology 51: 1106-1115.

Higgins, E. T., 1996. Knowledge activation: accessibility, applicability, and salience. Social psychology: Handbook of basic principles. E. T. Higgins and A. W. Kruglanski. NY, The Guilford Press.

Jacowitz, K. E. and D. Kahneman, 1995. Measures of anchoring in estimation tasks, Personality and Social Psychology Bulletin 21: 1161-1166.

Jørgensen, M., 2004. Top-down and bottom-up expert estimation of software development effort, Information and Software Technology 46(1): 3-16.

Jørgensen, M. and S. Grimstad, 2008. How to Avoid Impact from Irrelevant and Misleading Information When Estimating Software Development Effort, IEEE Software In Press.

Jørgensen, M., K. H. Teigen and K. Moløkken, 2004. Better sure than safe? Over-confidence in judgement based software development effort prediction intervals, Journal of Systems and Software 70(1-2): 79-93.

Koehler, D. and N. Harvey, Eds, 2004. Blackwell Handbook of Judgment and Decision Making. Blackwell Handbooks of Experimental Psychology, Blackwell Publishing.

Kruglanski, A. W. and T. Freund, 1983. The freezing and unfreezing of lay-inferences: Effects on impressional primacy, ethnic stereotyping and numerical anchoring, Journal of Experimental Social Pscyhology 19(5): 448-468.

Kusters, R. J., M. J. I. M. Genuchten and F. J. Heemstra, 1990. Are software cost-estimation models accurate?, Information and Software Technology 32(3): 187-190.

Lederer, A. L. and J. Prasad, 1995. Causes of inaccurate software development cost estimates, Journal of Systems and Software 31(2): 125-134.

Moløkken, K. and M. Jørgensen, 2003. A review of software surveys on software effort estimation. International Symposium on Empirical Software Engineering, Rome, Italy, Simula Res. Lab. Lysaker Norway.

Mussweiler, T., 2003. Comparison processes in social judgment: Mechanisms and consequences, Psychological Review 110: 472-489.

Ropponen, J. and K. Lyytinen, 1997. Can software risk management improve system development: an exploratory study, European Journal of Information Systems 6(1): 41-50.

Stapel, D. and W. Koomen, 1998. Interpretation versus Reference Framing: Assimilation and Contrast Effects in the Organizational Domain, Organization Behaviour and Human Decision Processes 76(2): 132-148.

Thomas, K. E., S. J. Handley and S. E. Newstead, 2007. The role of prior task experience in temporal misestimation, Quarterly Journal of Experimental Psychology 60(2): 230-240.

Tversky, A. and D. Kahneman, 1974. Judgment under uncertainty: Heuristics and biases, Science 185: 1124-1131.