Sequence effects in the estimation of software development effort

Magne Jørgensen, Simula Metropolitan & Oslo Metropolitan[1]

Torleif Halkjelsvik, Simula Metropolitan

Abstract: Currently, little is known about how much the sequence in which software development tasks or projects are estimated affects judgment-based effort estimates. To gain more knowledge, we examined estimation sequence effects in two experiments. In the first experiment, 362 software professionals estimated the effort of three large tasks of similar sizes, whereas in the second experiment 104 software professionals estimated the effort of four large and five small tasks. The sequence of the tasks was randomised in both experiments. The first experiment, with tasks of similar size, showed a mean increase of 10% from the first to the second and a 3% increase from the second to the third estimate. The second experiment showed that estimating a larger task after a smaller one led to a mean decrease in the estimate of 24%, and that estimating a smaller task after a larger one led to a mean increase of 25%. There was no statistically significant reduction in the sequence effect with higher competence. We conclude that more awareness about how the estimation sequence affects the estimates may reduce potentially harmful estimation biases. In particular, it may reduce the likelihood of a bias towards too low effort estimates.

Keywords: effort estimation, human judgment, sequence effect, software development

1. Introduction

Frequently, software professionals are required to estimate the effort needed to complete a set of software tasks. The set of software tasks to be estimated can be a larger set of activities or deliveries described in a project work breakdown structure (WBS) [1], a set of tasks to be completed for the next software release [2], or a set of user stories to be estimated in an agile planning poker session [3]. When estimating the effort required to complete a set of tasks, which is typically done through expert judgment [4], decisions on the sequence in which they are estimated must be made.

[1] Corresponding author: Magne Jørgensen, magnej@simula.no, Simula Metropolitan, P.O.Box 134, 1325 Lysaker, Norway

The research questions addressed in this paper are related to how much, if at all, the estimation sequences affect, and potentially bias, the effort estimates. Are there, for example, sequences that bias effort estimates towards lower estimates, and, does it matter whether an estimate of a larger task is given just after a smaller or just after a larger task? The research questions are, amongst others, motivated by the empirical findings documenting sequences effects leading to biased judgments in other domains. We know, for example, that people's judgments may be affected by judgments made immediately before [5], and that prior judgment may act as an "anchor" making the prior and subsequent judgements more similar [6]. The research questions are also motivated by the belief that knowing more about the presence and nature of sequence effects in software development effort estimation may be useful to guide the estimation work. In particular, it may be useful to guide the estimation sequences to avoid biases towards strong under-estimation, given the tendency towards too low effort estimates in software development [7].

We are not aware of any guidelines or reported experience, and there is only limited empirical research on whether some effort estimation sequences are better than others in terms of avoiding estimation bias and lower risk of too low effort estimates.

The remaining part of this paper presents related work on the sequence effects in judgments, including relevant theories and explanations regarding potential underlying mechanisms (Section 2). Based on the related work, we formulate three research hypotheses (Section 3). The following sections (Sections 4 and 5) present two empirical studies addressing the hypotheses. Finally, we discuss the results, including their limitations and implications (Section 6) and conclude (Section 7).

2. Related Work on Sequence Effects

Sequence effects in people's judgments have been studied for decades, and there are numerous competing explanations and descriptive models. Research studies suggest that sequence effects may occur because prior judgments are used as *references* in comparison-based judgments, because the prior judgment influences what is *recalled* of experiences, and because the prior judgments influence the *judgment process* itself [8].

The nature of the sequence effects varies between contexts. Professional judges' evaluations of gymnasts' performance in the 2004 Olympic Games were, on average, higher when the evaluated gymnasts performed after a gymnast with high scores and lower when they performed after a gymnast with relatively low scores [9]. The opposite sequence effect was discovered for oral examinations, where oral examinations conducted after a student who got a good grade led to lower grades, whereas examinations after a student who got poor grades led to higher grades [10]. The first effect, when a judgment is biased towards becoming more similar to the previous one, is called an *assimilation* effect, while the second effect, when a judgment conducted immediately after a previous judgment is biased towards becoming more different than the previous one is called a *contrast* effect [8, 11]. The domination of the contrast or assimilation effects seem to depend on the context, and there may be more than one mechanism involved in both types of effects [12, 13].

Notice that assimilation and contrast effects, which are the sequence effects we focus on in this paper, are only terms used to describe two types of sequence effects. They do not aim to explain or present the underlying mechanisms for the observed bias towards judgements becoming more similar or more different when judgments are completed in sequence. For that purpose, we need supporting theories and mechanisms. Below, we present briefly a selection of what we believe are the most prominent mechanisms and theories underlying sequence effects.

2.1 Contrast Effects: Time Order Error and Feature Matching Theory

The first report on sequence effects on judgments is by Gustav Fechner in 1860 [14]. He observed that when people tried to judge the heavier of two objects by lifting them in sequence, the weight of the one lifted later tended to be more frequently assessed as heavier when actually being lighter or the same weight. He described this phenomenon as a 'remaining effect' (Nachdauer) of the first on the second judgment. This type of temporal sequence effect, which in psychophysics is called a time order error, has been observed in several experiments on the judgment of physical stimuli such as loudness, brightness, taste, duration, and number of objects [15]. Most often, similar to the initial study by Fechner, the second object tends to be overestimated relative to the first. This effect means, amongst other things, that when comparing the weight of two objects of the same weight, people tended to think that the second one was the heaviest. The results on the time order error are robust, but there are competing explanations.

Psychophysical theories explain the effect, amongst others, in terms of the differential weighting of successive stimuli and as the blending of the prior stimulus into the current one [15].

A theory from psychology that explains the contrast effects is the feature matching theory [16]. This theory posits that when comparing a target with a reference, we tend to focus more on identifying the unique characteristics of the target and less on those of the reference. Assume, for example, that we first estimate the effort required to complete task A and then the effort required to complete task B, and that tasks A and B are similar in the effort required. According to the feature matching theory, people will perceive task B (the target) to have more unique features than task A (the reference) and are more likely to judge it to require more effort.

2.2 Assimilation effects: Anchoring

The anchoring effect is one of the most robust and best-documented cognitive biases. It occurs when judgments are preceded by a presented or self-generated value (the anchor value). The anchor value tends to affect the subsequent judgement so that it gets more similar to the anchor value, and this way it enables an assimilation type of sequence effect. Anchoring effects are known to be present even when participants are explicitly informed about their presence or when it is obvious that the anchoring information is irrelevant [17].

Researchers have established the presence of the anchoring effect using many different types of judgments and anchors [18], including those found in important real-world judgements such as sentencing decisions of judges in criminal trials [19], and estimates of property values by real estate agents [20]. Anchor values such as the presentation of a very low effort estimate suggested by a technically incompetent client before requesting an effort estimate have also been documented to affect software development effort estimates [21-24]. This suggests that, more than the content of the anchoring object or task, the value (the number) itself is essential to create anchoring effects [25]. The results reported in [9, Study 1, 26] suggest that the assimilation effect owing to anchoring does not arise from the immediately preceding judgment only, but may also be affected by judgments earlier than that.
Similar to the time order error, the anchoring effect results are robust, but with little agreement regarding its explanation. The *scale distortion theory* reported by Frederick and

Mochon in [27], proposes that the anchoring effect occurs because of a shift in the interpretation and the use of response scale. According to this theory, a low effort estimate (low number) of a previous small task will influence people to feel that high estimates (high numbers) for a subsequent larger task are higher than normal because they are large compared to the previous estimate. Consequently, they will select a lower estimate (lower number) compared to situations where the preceding and current tasks are similar in size. For example, if a previous smaller task is estimated to require five hours, then an estimate of 100 hours of the subsequent larger task seems higher than otherwise and a lower number (such as 80 hours) may instead be provided as the effort estimate. If the preceding task were much larger than the one to be estimated, the opposite would occur. In that case, a low estimate (low number) seems very low, and higher effort estimates than otherwise are selected for the task. The scale distortion theory consequently predicts that estimating a larger task after a smaller one would lead to lower estimates and estimating a smaller task after a larger one would lead to higher estimates.

Strack and Mussweiler's *selective accessibility model*, see [28], states that the anchor value makes knowledge consistent with the anchor value more accessible, and, consequently, more likely to be used in the subsequent judgement through a process of confirmatory hypothesis testing. For example, estimating the effort necessary for a small task makes the knowledge about smaller tasks more accessible because this knowledge has been used (activated) recently. Therefore, the knowledge about smaller tasks is more likely to be used when estimating a subsequent larger task and influences its effort estimate downwards towards that of smaller tasks.

The initial explanation of the anchoring effect was that people used an *anchoring-and-adjustment heuristic*, i.e. they started with the anchor value and adjusted, typically insufficiently, for the difference between the anchor and the target value. In this manner, i.e. by insufficient adjustments away from the anchor, an assimilation type of sequence effect is created. This explanation has little support for externally induced anchors; however, it receives some support for some types of self-generated anchors [29].

2.3 Judgment-based Effort Estimation

It is known from several studies that software developers' effort estimates are not always fully rational and unbiased, and that they may be affected by irrelevant and misleading information presented before the estimation work, see for example [30, 31]. Additionally, studies from psychology suggest that previous judgments tend to be used as references for the judgment following it, and consequently affect it, see for example [5, 32]. For example, in a set of psychology experiments, participants were asked to estimate the effort necessary in different tasks, such as building a toy castle or solving the tower of Hanoi problem [25, 33, 34]. The experiments observed consistently higher effort estimates when a task was preceded by the estimation and completion of a larger task and lower estimates when the task was preceded by the estimation and completion of smaller tasks. Similarly there are software development results suggesting that effort estimates of a medium large task tend to be higher when produced immediately after the estimation of a larger task instead of a smaller one [35], and that effort estimates of the second of two similarly sized tasks tend to be higher [36].

The relation between sequence effects and task completion competence is not clear. While the results reported in [37-39] suggest that sequence effects are smaller when the task competence is high, the results reported in [35, 40, 41] suggest no or only modest connections between competence and sequence effects.

3. Hypotheses

The hypotheses to be tested, motivated by the related work in Section 2, are as follows:

H1: Estimating the effort of a software development task after a *similarly sized* task tends to increase its estimate, i.e. produce an effort-increasing contrast effect.

H2: Estimating the effort of a software development task after a substantially *differently sized* task tends to make the estimate more similar to the previous task, i.e. produce an assimilation effect.

H3: The contrast and assimilation effects described in hypotheses H1 and H2 increase in strength with decreased task competence.

Notice that Hypothesis H2 implies that estimating the effort necessary for a software development task after a *smaller* task tends to decrease its estimate, whereas estimating the task after a *larger* task tends to increase its estimate.

The two experiments that are part of this paper test the above hypotheses. In addition, they enable us to evaluate the validity of and extend the prior results on sequence effects in software development effort estimation contexts, as reported in the two studies [35, 36]. These two prior studies analysed the effort estimation sequences of only two tasks and were conducted with relatively few observations. The studies reported in this paper extend those studies with longer sequences of effort estimates and with larger sample sizes. In addition, the results reported in this paper include findings on to what extent task specific competence, and not only general software development competence, moderates the sequence effects.

4. Study A: Estimation of similarly sized tasks

4.1 Participants

We recruited 373 software professionals from five East European and seven Asian outsourcing companies.[2] All the participants were required to have at least a bachelor's degree in computer science or similar study programmes and at least six months of software development experience. 41% of the participants had a master's and 59% a bachelor's degree. Their degrees were mainly in computer science or computer engineering (72%) but were also in information systems, management or economics (7%), mathematics or physics (6%), electronics or engineering (5%), and other areas (10%). The participants had a mean age of 26 years and a mean of three years of software development experience. Thirty-nine per cent of the participants had experience as project managers. Females comprised 14% of the participants. The companies were paid their regular hourly rate for the work of their employees.

4.2 Study design

The participants were informed about the content of the study and that its main purpose was to gain knowledge about how to improve effort estimation work. They were *not* informed that we would analyse sequence effects in their effort estimates. The participants were provided with a leaflet with instructions and information about the effort estimation tasks, with the instruction not to look at the next page of the leaflet before the task on the current page was

[2] Our analyses are based on the same data set; however, without overlap in analyses, the one reported in [30].

fully completed. They were also informed that their answers were strictly confidential, and nobody would be able to identify them or their company from the research reports. The first part of the leaflet instructed them to report information about themselves, such as age, gender, and self-assessed software development competence. In the second part of the leaflet, they were required to estimate the effort they would need, assuming that they performed all the work themselves and used the tools and programming languages they knew best to develop three different software systems:[3]

- Photo: Adding photo functionality to an existing platform for e-dating.
- Ticket: Adding bulk ordering functionality to an existing ticket ordering system.
- Nurse: Development of a simple time shift system for hospital nurses.

The specified systems vary in the type of functionality to be developed. Based on the mean and median values of the effort estimates provided by other developers on the same tasks in prior experiments [40, 42], we expected that the systems would result in similar estimates with median values of approximately 100 work-hours. The sequence of presenting the systems was randomised for each software professional.

Of the 373 software professionals, 11 did not provide complete responses, thereby, leading to a total of 362 participants and 362 x 3 = 1086 estimates. The lack of complete responses was partly because of the respondents' perceived lack of competence in at least one of the tasks and partly as a result of them forgetting to complete an estimation task.

4.3 Results

Analogous to the previous estimation experience with the tasks, the mean and median effort estimates of the three tasks were quite similar, as seen in Table 1.

Table 1: Mean and median estimates of the tasks

Task (n=362)	**Mean** (hours)	**Median** (hours)
Photo	168	80

[3] The requirement specifications of the systems can be provided by the corresponding author on request. There were variations in the instructions regarding the estimation work used to test cultural differences in estimation biases. These variations were not about the actual work to be, and given the randomized sequence and treatments, they were not a threat to the validity of the analyses of the direction of the sequence effects.

Ticket	154	100
Nurse	165	98

For the purpose of evaluating the presence of a sequence effect in the effort estimates, we ran a linear mixed model with the logarithm of the estimated effort (lnEst) as the response variable, company and participant (nested in company) as random intercepts, and estimation task (nurse, photo, ticket) and sequence (whether a task was estimated as first, second, or third) as dummy-coded fixed variables. The logarithm (lnEst) was used because the distributions of the original effort estimates were strongly right-skewed. The log-transformation made the estimation variables close to normally distributed. The use of the logarithm had as a consequence that, when back-transforming the mean estimates to the original scale, we obtained the geometric instead of the arithmetic mean. The variance estimation was based on the restricted maximum likelihood, and the tests of fixed effects used the Kenward-Roger degrees of freedom approximation. The model produced close to normally distributed conditional residuals.

Table 2 lists the model parameters, whereas Table 3 lists a test of the sequence effect and the back-transformed mean values of lnEst.

Table 2: Mixed model parameters

Fixed effects	**Variable**	**Categories**	**Coefficient**	**95% CI**
	Intercept		4.57	[4.41; 4.73]
	Task (Nurse is the reference)	Photo	-0.10	[-0.15; -0.05]
		Ticket	0.06	[0.01; 0.11]
	Sequence (The third task is the reference)	First	-0.07	[-0.12; -0.02]
		Second	0.02	[-0.03; 0.07]
Random effects	**Variable**	**Variance**	**Percent of total variance**	
	Company	0.04	4%	
	Participant	0.64	62%	
	Residual	0.35	34%	

Table 3: Test of sequence effects

Item	**Tests**

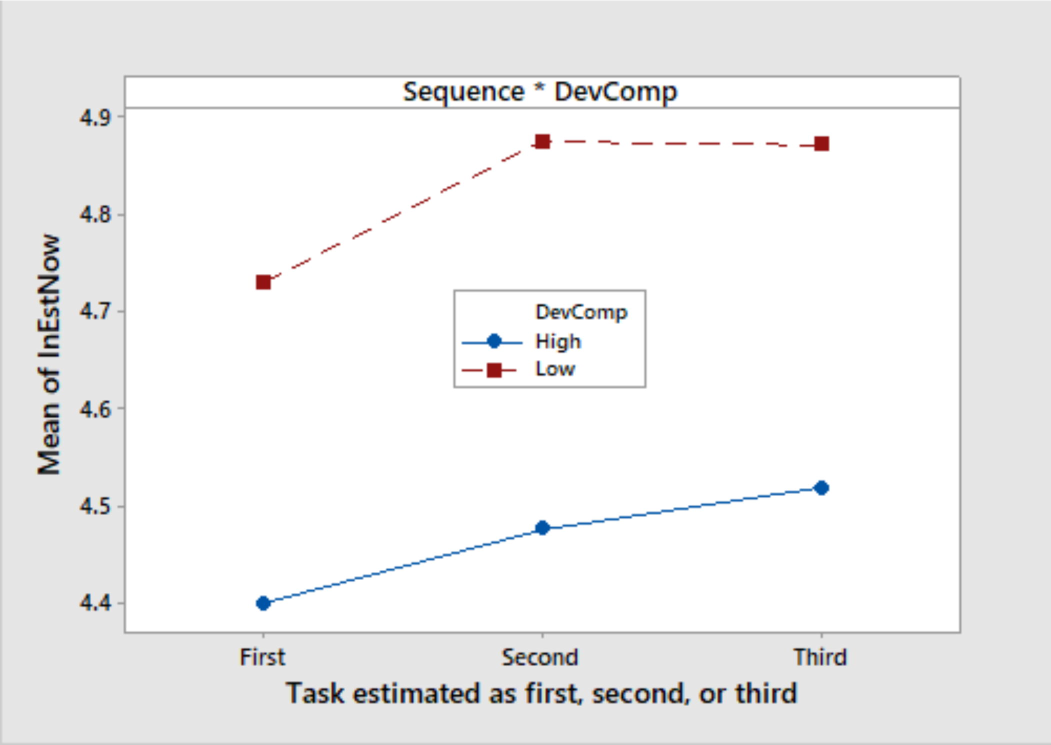
Sequence	F-value 4.23, p-value = 0.015
Mean estimates of the first, second, and third tasks back-transformed from lnEst.[4]	First task: 89 hours (lnEst=4.49) Second task: 98 hours (lnEst=4.59) Third task: 101 hours (lnEst=4.62)

As seen in Table 3, the sequence of the tasks had a statistically significant effect on the estimated effort. There was a 10% increase (from 89 to 98 hours) in estimates from the first to the second task, and a 3% increase (from 98 to 101 hours) from the second to the third task. Both differences in estimates were in the expected direction and support hypothesis H1, i.e., that estimating the effort of a software development task after a *similarly sized* task tends to increase its estimate. The results also suggest that the contrast effect may be mainly from the first to the second estimate, i.e., an initial sequence effect, and not so much from the second to the third task.

To test the effect of task competence on the sequence effect (hypothesis H3), we extended the linear mixed model with the variable self-assessed competence (DevComp) and the interaction between the Sequence and DevComp (Sequence x DevComp). DevComp was categorised as 'High' if the developer assessed themselves as having 'Very high' or 'High' development skill (n=277) and as 'Low' if having 'Average', 'Low', or 'Very low' development skill (n=83). Low self-assessed competence did have a statistically significant (F-value of 8.59, p=0.004) main effect on the effort estimates; however, the interaction effect between competence and sequence was low and not statistically significant (F-value of 0.22, p=0.80), i.e., the sequence effect was not very different for the two competence groups. This lack of interaction effect is visualized by the nearly parallel lines in Figure 1. Consequently, the data does not support hypothesis H3. Those with high competence seem to be affected similarly by the effort estimation sequence compared to those with less competence.

Figure 1: Interaction plot of lnEst

[4] The back-transformed means (geometric means) will be close to the *median* values (Table 1) in the original effort estimation distribution if the log-transformed effort distribution is close to normal, which is the case in this study.



Self-assessed *general* competence in software development may not be a good indicator of competence in solving the *particular* tasks to be estimated. It may also be that the inclusion of software professionals from different cultures, possibly with different self-assessment standards, reduced our ability to find competence effects. In study B, we decided to include a more task-specific competence measure and software professionals from only one region (East Europe).

5. Study B: Estimation of differently sized tasks

5.1 Participants

Two software companies located in two different East European countries (companies A and B) were contacted and asked to provide Java developers with at least half a year of Java development experience to participate in the study. To include a variety of competence levels and reduce the risk of a non-representative sample of developers, we requested a mixture of ‘junior’, ‘intermediate’, and ‘senior’ Java developers, which were the categories used by the companies to differentiate hourly payment rates according to skills and experience. The companies were offered their regular hourly fees as payment. The hourly payment rate was approximately 20% higher for an intermediate than for a junior and 20% higher for a senior than for an intermediate developer.

Both companies accepted the request and offered 104 software developers in total for the work, of which 27% were categorised as junior, 43% as intermediate, and 30% as senior developers. Seven per cent of the developers were female. The mean length of experience as software professionals was 7.5 years (ranging from 0.7 to 27 years), and the mean length of experience as Java programmers was 5.8 years (ranging from 0.5 to 18 years). Seventy per cent of the developers had been responsible for effort estimates on software projects at least once, and 96% for the effort estimation of their own work at least once.

5.2 Study design

The participants were informed about the study's general purpose, but not that we would study the sequence effects of their estimates. They were also informed that the data on the individual level would be treated as confidential information, and that only aggregated and anonymised information would be published. The participants chose when to start the study, within a time frame of approximately two weeks. At the beginning of the study, they were instructed not to get interrupted by other tasks before finishing the work. All the participants provided information about their experience and role and estimated the effort required to complete the nine software development tasks using the Java programming language in their preferred programming environment. We categorised four of the tasks as 'large', and five as 'small' (Table 4 includes a description of the tasks). The survey tool Qualtrics (www.qualtrics.com) was used for data collection, and the sequence of tasks was randomised for each participant.

Table 4: Characteristics of tasks

Task	Size category	Description
A	Large	Development of a web-system with functionality enabling search, display, and import of information about scientific articles.
B	Large	Development of a database with functionality for inserting, updating, and removing data about empirical studies and with links to other databases.
C	Large	Development of a web-based system that graphically displays selected connections on a world map such as the number of outsourced projects between different countries.
D	Large	Development of a standalone desktop system with rule-based support for a selection of jobbing shoes.
E	Small	Adding functionality to an existing ATM application. Code to be updated, and design diagram included as task documentation.
F	Small	Adding functionality to an existing program for a coffee vending machine. Code to be updated, and design diagram included as task documentation.
G	Small	Modifying a laser controller program. Code to be updated, and design diagram included as task documentation.
H	Small	Development of a program that lists the directory content of a specified root directory.
I	Small	Fixing a bug and extending the functionality of a lab-order system of a health-care related software.

Based on the mean and median values of previous estimates provided by software professionals on the same tasks [43, 44], we expected that both the large and the small tasks would not be very different in size and that the large tasks would be substantially larger than the smaller ones.

For the purpose of testing hypothesis H3 with a measure of task-specific competence, we asked the developers about how confident they were in knowing how to solve the estimated task. The responses were placed on a scale from 1 (No idea what to do) to 6 (Know exactly what to do). The questions about task knowledge were asked immediately after the estimation of each task, i.e., nine times for each developer.

5.3 Results

As seen in Table 5, the differences in mean and median effort estimates of the large and the small tasks were substantial. The tasks belonging to the same size category were not as similar in size as in study A. They were, however, considered to be similar enough to enable an examination of the potential contrast effects.

Table 5: Mean and medians of the estimates

Task	Size category	Mean estimate	Median estimate
A	Large	112 hours	62 hours
B	Large	234 hours	158 hours
C	Large	134 hours	40 hours
D	Large	125 hours	56 hours
E	Small	451 minutes	155 minutes
F	Small	321 minutes	120 minutes
G	Small	179 minutes	60 minutes
H	Small	291 minutes	90 minutes
I	Small	744 minutes	390 minutes

As indicated by the substantially higher mean than median effort estimates, the underlying distributions of estimates are strongly right-skewed. Therefore, similar to study A, we decided to use the log-transformed estimates (lnEst) in our statistical analyses. To simplify the interpretation of the results, we performed two separate main analyses: One analysis with the estimates of the larger tasks and a second with the estimates of the smaller tasks as the outcome variable.

We used linear mixed models with participant as a random effect variable, and company, task and size category of the the previous task (small or large) as fixed effect variables. The conditional residuals of both models were found to be close to normally distributed. The model parameters for the two models are listed in Tables 6 and 7. Table 8 lists the results of the tests of the sequence effects using the models and the back-transformed mean values.

Table 6: Linear mixed model parameters for larger tasks

Fixed effects	**Variable**	**Categories**	**Coefficient**	**95% CI**
	Intercept		8.32	[8.09; 8.55]
	Company (company B is the reference)	Company A	0.36	[0.13; 0.59]
	Task (task D is the reference)	Task A Task B Task C	-0.22 0.61 -0.20	[-0.34; -0.10] [0.50; 0.74] [-0.33; -0.07]
	Previous task category (small size of the previous task is the reference)	Large	0.13	[0.05; 0.21]
Random effect	**Variable**	**Variance**	**Percent of total variance**	
	Participant	1.22	73%	
	Residual	0.45	27%	

Table 7: Linear mixed model analysis for smaller tasks

Fixed effects	**Variable**	**Categories**	**Coefficient**	**95% CI**
	Intercept		5.06	[4.86; 5.26]
	Company (company B is the reference)	Company A	-0.24	[-0.45; -0.04]
	Task (task I is the reference)	Task E Task F Task G Task H	0.22 -0.07 -0.81 -0.37	[0.08; 0.36] [-0.22; 0.07] [-0.95; -0.67] [-0.52; -0.22]
	Previous task category (small size of the previous task is the reference)	Large	0.11	[0.03; 0.18]
Random effect	**Variable**	**Variance**	**Percent of total variance**	
	Participant	0.92	60%	
	Residual	0.61	40%	

Table 8: Test of the sequence effect for larger and smaller tasks

Item	Large tasks	Small tasks
Previous task category (large or small)	F-value 11.3, p-value = 0.001	F-value 7.78, p-value 0.006
Mean estimates, back-transformed from lnEst, dependent on whether the previous task was large or small.	Large: 78 hours (lnEst=8.45) Small: 59 hours (lnEst=8.18) Estimates of the large tasks were on average 19 hours (24%) lower when estimated after a small task compared to when estimated after a large task.	Large: 176 minutes (lnEst=5.17) Small: 141 minutes (lnEst=4.95) Estimates of the small tasks were on average 35 minutes (25%) higher when estimated after a large task compared to when estimated after a small task.

As seen in Table 8, the size category of the previous task affected the effort estimates of both the large (24% difference) and the small tasks (25% difference), i.e., there were substantial and statistically significant sequence effects. The differences in estimates dependent on whether the previous task was large or small do not, however, document to what extent the differences belong to contrast effects, assimilation effects, or combinations of the effects.

The contrast effect, assuming that it has the effect hypothesised in H1, increases the estimates of a large task estimated after a large task, while the assimilation effect decreases the estimates of a large task estimated after a small task. In other words, both effects contribute to the observed differences between a large task estimated after a small or large task. This result means that it is not possible to know to what extent the sequence effects on the estimates of the *large* tasks can be attributed to an assimilation effect, a contrast effect, or a combination of both effects.

For the *small* tasks, however, we can deduce that the assimilation effect must have been stronger than the contrast effect. If the contrast effect had been the strongest, the difference in estimates would be in the opposite direction, i.e., estimating a small task after a small one would provide higher estimates than estimating a small task after a large one. This result provides support to hypothesis H2.

A more direct analysis on how much of the sequence effect it is reasonable to attribute to contrast and how much to attribute to assimilation effects, although only valid for the initial sequence effect, is an analysis including only the *two first* effort estimates provided by each developer, i.e., the estimation sequence with the first estimates not affected by any immediate sequence effect and the second estimate preceded only by an estimate of only one small or only one large task.

We ran a linear mixed effect analysis with the developer as the random factor and the company, task, and previous task category as fixed factors. The variable representing the size category of the previous task now included the values first (when a task was estimated as first, i.e., no previous task), large (when a task was estimated after a large task), and small (when a task was estimated after a small task).

Table 9 lists the resulting back-transformed mean estimates. All differences in the estimates were in the hypothesised directions and close to (F-value 2.72, p-value 0.07 for the large tasks) or just (F-value 3.34, p-value 0.04 for the small tasks) statistically significant. These results together provide further support for our hypotheses H1 and H2, i.e., there are assimilation effects for differently sized tasks and effort-increasing contrast effects for similarly sized tasks. Furthermore, the presence of large contrast effects, despite that the tasks of the same size category were not as similar in size as in Study A, suggest that the tasks do not need to be very similar in size to experience contrast effects.

Table 9: Comparison of mean estimates (back-transformed from lnEst) of the first two estimates

Mean estimates	**When estimated as first**	**When estimated after a large task**	**When estimated after a small task**	**Contrast effect (same sized category)**	**Assimilation effect (different sized category)**
Large tasks	55 hours (lnEst=8.11)	83 hours (lnEst=8.51)	45 hours (lnEst=7.90)	42 hours (83-55 hours)	10 hours (55-45 hours)
Small tasks	114 minutes (lnEst=4.74)	179 minutes (lnEst=5.19)	151 minutes (lnEst=5.02)	37 minutes (151-114 minutes)	65 minutes (179-114 minutes)

To test the moderating effect of competence (H3), the developers’ assessments about how much they knew about how to solve each of the tasks (DevComp) were grouped into two categories 'low' (scores 1–3) and 'high' (scores 4–6). This task competence variable ('low' or 'high') and its interaction with the size category ('small' and 'large') from the previous task was added to the linear mixed models as a fixed main effect and a fixed interaction effect, respectively.

The analysis indicates that the interaction variable is neither significant for the large tasks (F-value 1.07, p-value 0.30) nor the small tasks (F-value 1.18, p-value 0.28). The resulting interaction plots illustrated in Figures 2 (large tasks) and 3 (small tasks) show that the lines of those with low (solid lines) and high (dotted lines) competence have a similar slope, i.e. there is no clear interaction between the sequence effect and competence. In total, the effect of the perceived level of knowledge on how to solve a task on the sequence effect seems to be low or none at all, and, consequently, our results do not support hypothesis H3.

Figure 2: Estimated effort dependent on size category of previous task and developer competence (large tasks)

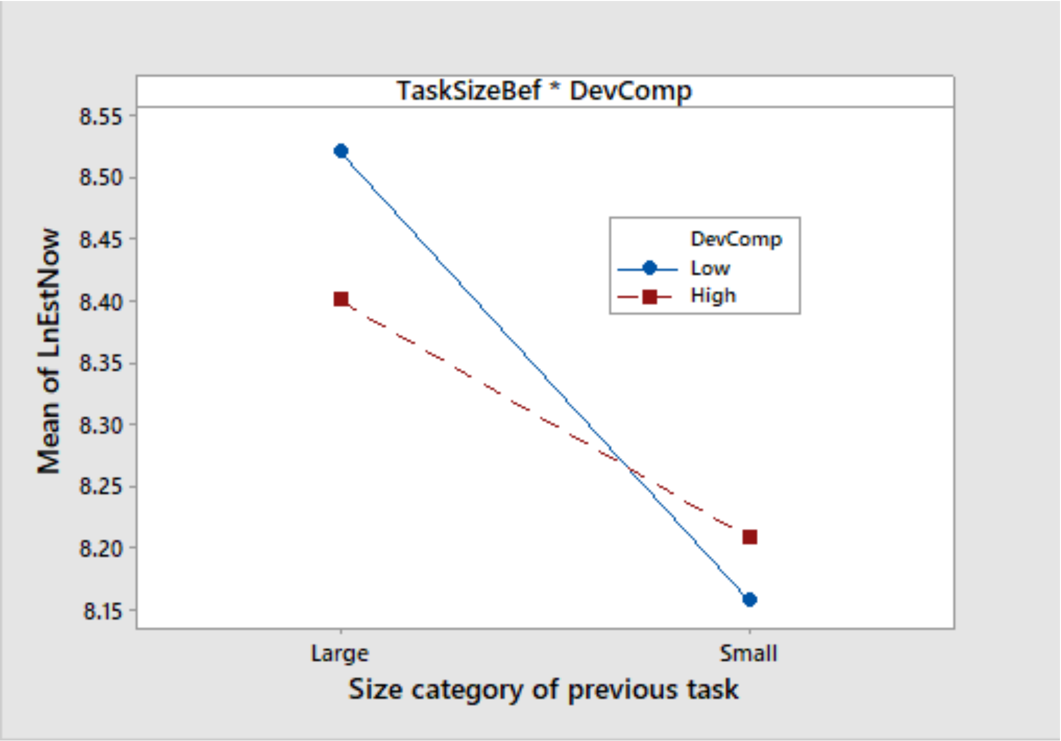
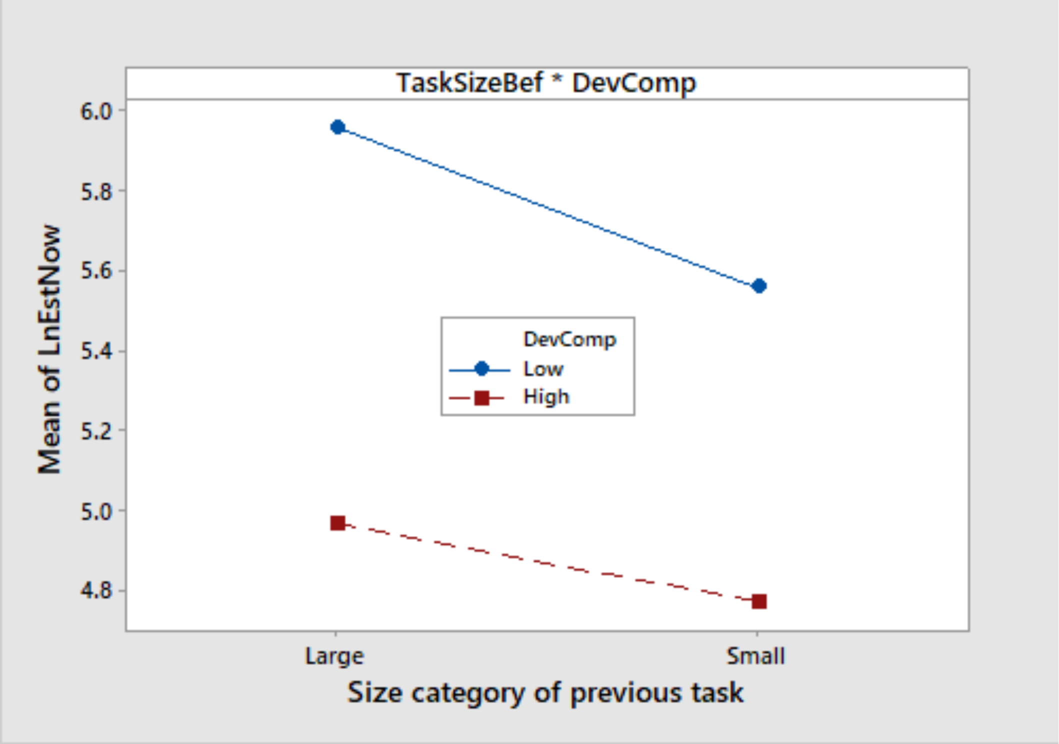


Figure 3: Estimated effort dependent on size category of previous task and developer competence (small tasks)



6. Discussion

The two experiments reported in this paper observe effort-increasing contrast effects for effort estimates of tasks of similar sizes, and assimilation effects for tasks of substantially different sizes, i.e., they provide support for our hypotheses H1 and H2. They did not observe reduced sequence effects with general or task-specific, self-assessed software-development competence, i.e., they provide no support for our hypothesis H3.

Our results on the *contrast* effect share similarities with results from the time order error studies in psychophysics. As an illustration, consider a recent psychophysics study, where people estimated the number of dots present in two pictures [45, Study 3]. The two pictures were presented in random sequences, and they either had the same or a similar number of dots. That study reports an average of 8% increase in estimates when the number of dots in a picture was shown as the second compared to when shown as the first of the sequence, i.e., an increase in estimates similar to the 10% increase observed from the first to the second task in Study A. The contrast effect reported in [36, Study 2] is substantially larger than those reported above, with effort estimate increases of 25% and 33% from the first to the second of two similarly sized software development tasks. These two increases are more similar to the estimate increases reported in our Study B, where the increase from the first to second of two similarly sized tasks was 51% for the larger and 32% for the smaller tasks. The observed differences in contrast effects in our Studies A and B and prior studies, suggest that the effect sizes of the contrast effect may vary much from estimation context to context. According to [8, 11], and the proposed mechanisms for the contrast effects described in Section 2, there may be elements of the comparison conditions that make people emphasise differences that lead to contrast effects. These difference-inducing elements may be hidden in how the estimate request is formulated or in other parts of the estimation task. The nature and effect of these elements are currently not well understood, and in need of further research.

The *assimilation* effects reported for the two first tasks in our Study B were that the estimate of a larger task became 18% lower when estimated after a small task, and the estimate of a smaller task became 57% higher when estimated after a larger task. In the prior study on assimilation effects in software development effort estimation, see [35], it is reported that a medium-large task was estimated to require as much as 105% more effort when estimated after a smaller compared to a larger task. This difference in result may partly be explained by that the design in [35] to some extent "doubles" the sequence effect compared to our study B, i.e., it compared estimates of a medium-large task biased towards too low with estimates of the same medium-large task biased towards too high estimates. When adjusting for this design difference, the results of our Study B may not be very different from those found in [35].

Our studies suggest that the contrast effects dominate when tasks are similar in size and that the assimilation effects dominate when the tasks differ much in size. Possibly, the mechanisms leading to both assimilation and contrast always exist, but their strength and ability to affect the effort estimates vary with the context. A context where we probably will not observe an assimilation effect on effort estimates is when tasks estimated in a sequence are very similar in size. In that context, the task efforts are already very similar, an assimilation effect cannot be observed, and only contrast effects will be observable [45, 46]. The assimilation effect has, accordingly, the potential to dominate the sequence effect on effort estimates mainly in contexts with large differences in the sizes of tasks estimated in a sequence.

There may be diminishing sequence effects as the sequences get longer, such as the decrease in the contrast effect from the second to the third task reported in Study A. Diminishing sequence effects may, for example, be present when the effort estimation of a large task is preceded by the estimates of one large and one small task. In this case, the estimate of the large task preceding the small task may moderate the sequence effect (assimilation effect) of the small task on the large task. There may, however, also be longer sequences strengthening the sequence effect. This may, for example, happen when the effort estimate for a large task is preceded by the estimation of many smaller tasks, potentially strengthening the assimilation effect towards lower estimates of the large task. We tried to add analyses of different sub-sequences of estimates, such as the one exemplified above. Unfortunately, the number of observations in each sub-sequence tended to be too low to enable robust analyses. The analysis of sub-sequences is an interesting topic for further research, requiring larger datasets or an experimental design tailored to compare specific estimation sequences.

We did not find a reduced sequence effect with increased competence. This observation may have been a result of that we measured the self-assessed level of competence and did not use more objective measures for competence. The analysis in [47], covering more than 2000 studies, suggests that a correlation between self-assessed competence and actual performance (indicating actual competence) typically is around 0.3. Applying this seemingly robust finding implies that we would have found a moderating effect from self-assessed competence, if the true effect of competence was large, but not if it was only moderately high. While we cannot exclude a moderating effect from competence based on our studies, we may, consequently, argue that it is not likely to be large.

Limitations

There are convincing reasons, amongst others based on prior studies, to believe that contrast and assimilation effects exist and that they can be substantial. Our results are consequently not extraordinary or contradictory to the main findings from the domains of psychophysics and human judgment. A limitation of our results, shared with that of the results of the prior studies, is the current poor or diverging understanding of why and when to expect contrast and assimilation effects, and how large they will be in different contexts. As pointed out earlier in this section, the sequence effects vary considerably from context to context, and we are unable to explain why.

The poor understanding of why and when we can expect sequences effects and how large they will be also mean that we know little about how much the sequence effect decreases with increased time between the estimates. Research results on sequence effects, including those reported in this study, typically study judgments provided within minutes (or seconds) of each other, and we are unsure about what will happen in situations such as those with hours or a full workday in-between the effort estimates. Furthermore, we do not know what will happen when tasks get even more different in size than those in our study. Finally, the poor understanding of the mechanisms, effect sizes, and context dependency mean that we should be careful about generalising our results to other contexts. There is a need for further examination of contrast and assimilation effects in more diverse software development effort estimation contexts.

In Study B, we examined a longer task estimation sequence that included both contrast and assimilation effects. This approach was useful to assess the combined effect of contrast and assimilation effects when estimating a set of differently sized tasks. However, it also led to challenges related to knowing how much to attribute to each of the effects and whether there was a decreased sequence effect from the initial to the later estimates. More studies are required to address this limitation using research designs better suited for this purpose.

We argue that we would have found a moderating effect of increased competence if this effect was large, assuming a correlation between self-assessed and actual competence as reported in other studies. To find out how large this moderating effect is, e.g., whether there is none, a small, or a medium-large effect, we need studies with more objective measures of task performance or competence.

The participating software professionals in our two studies were from offshoring companies in Eastern Europe and Asia. While we have no reasons to believe that effort estimation biases, such as the sequence effects, are substantially higher or lower in some geographical regions (see our study reported in [30] for more on this issue), there may be elements in how estimation requests or task descriptions are formulated that are differently interpreted in different cultures. The replication of the sequence effect in other regions, and with other task formulations, would, consequently, add robustness to the results.

There are types of sequence effects not examined in this paper. One potentially interesting sequence effect is to what extent it makes a difference on subsequent estimates whether a developer starts with a low or high estimate on the first task. We recommend this as a topic for further research.

Implications for practice

Better awareness of the estimation sequence effects may reduce estimation biases and improve the realism of the effort estimates in software development. In particular, to reduce the risk of effort over-runs, it may be important to avoid estimating a large task, user story or project immediately after estimating one or more much smaller tasks, user stories, or projects. The risk of underestimating the largest tasks may, for example, be reduced by starting the estimation with those tasks, and not estimating them in a sequence with a mix of smaller and larger tasks. It may also be unfortunate to estimate a small task immediately after a large one. This sequence may bias the effort estimate of the small task towards too high effort estimates. Estimates which are too high may have negative consequences, such as decreasing the development productivity [48].

When decomposing a software project or software product release into tasks, the more similarly sized these tasks are, the less an assimilation effect is likely to be created. Assuming that the contrast effects are typically weaker than the assimilation effects, a result which receives some support in our studies, the effort estimates may be exposed to less biasing sequence effects.

The relative estimation method often used in agile projects, see [3], may also benefit from better knowledge about sequence effects, particularly about the assimilation effect. When applying relative estimation, it is recommended to identify and estimate the effort or size of one or more baseline (reference) tasks or user stories. When a new task or user story arrives, its effort or size is compared with that of the baseline. Our results may be useful to guide the selection of a baseline task. Selecting a relatively small task as the baseline, together with the assimilation effect, would lead to underestimation of the largest tasks, while selecting a relatively large task as baseline would lead to overestimation of larger tasks. The selection of a medium-large baseline task may lead to least estimation bias, as before assuming that the assimilation effect typically is larger than the contrast effect, and be the better choice.

It is not likely that we can remove the sequence effects from judgment-based effort estimation. If we, for example, succeed in reducing the assimilation effect by estimating only tasks of similar size, the estimation sequence will be exposed to the contrast effect. The best we can do may be to raise the awareness that sequence effects are there and try to use sequences that avoid the most damaging effects of them.

Implications for research design

The first observations of sequence effects were used to improve the research design [14], i.e., to avoid sequence effects that confounded the main effects studied. The first observation of sequence effects had researchers counterbalance or randomise the sequence so that sometimes the heavier and sometimes the lighter weight was lifted first. Our results on the sequence effect support the importance of adjusting for sequence effects in research studies. We believe that a stronger awareness regarding the need to randomise or counterbalance the sequence of questions and tasks in software engineering surveys and experiments is sometimes essential to avoid being misled by sequence effects.

Sometimes, randomisation or counterbalancing is not possible, e.g., in observational studies. If there are systematic biases in the sequences, e.g., that one starts with the estimation of the simplest tasks, the researchers should at least discuss and possibly try to assess the likely influence of sequence effects on the claimed results. As far as possible, the adjustment should be based on the sequence effects found in similar contexts.

7. Conclusions

Our two experiments reproduce and extend the results of prior studies on sequence effects in effort estimation of software development, as reported in [35, 36].

When estimating the effort of a sequence of tasks of different and similar sizes, we observe a both contrast and assimilation effects. We found increased effort estimates when estimating tasks of similar size in a sequence, i.e., the domination of the contrast effect, and effort estimates pulled towards that of the previous task when estimating tasks with more different sizes, i.e., the domination of the assimilation effect. The observed sequence effects were present despite the tasks being considerably different in content and when estimated as part of longer estimation sequences. The sequence effects were not much reduced with higher competence when measuring competence as a self-assessed general or self-assessed task-specific software development competence.

The results suggest that sequence effects play a role in explaining effort estimation biases and errors in software development contexts. More awareness of how the sequence in which we estimate the effort of software development tasks affects the estimates may be important to avoid the most undesirable estimation sequences. One potentially harmful sequence to be avoided, particularly in contexts where there already is a tendency towards underestimation of effort, is to estimate the effort of a larger task immediately after a smaller one. This sequence is, because of the assimilation effect, likely to increase the risk of underestimating the effort of the larger task. Similarly, it is important to reduce the risk of underestimating effort by avoiding the estimation of a smaller task just after a larger one because underestimation is known to decrease productivity [48].

Future work on sequence effects should, we argue, focus on how much the sequence effect reduces with more time in-between the effort estimates, the size of the sequence effect in other software development effort contexts, and a better understanding of the underlying mechanisms.

References:

[1] Tausworthe, R.C., *The work breakdown structure in software project management.* Journal of Systems and Software, 1980. **1**(3): p. 181-186.

[2] Hill, J., L.C. Thomas, and D.E. Allen, *Experts' estimates of task durations in software development projects.* International Journal of Project Management, 2000. **18**(1): p. 13-21.

[3] Cohn, M., *Agile estimation*. 2006: Prentice Hall.

[4] Jørgensen, M., *Forecasting of software development work effort: evidence on expert judgement and formal models.* International Journal of Forecasting, 2007. **23**(3): p. 449-462.

[5] Sharif, M.A. and D.M. Oppenheimer, *The effect of relative encoding on memory-based judgments.* Journal of Psychological Science, 2016. **27**(8): p. 1136-1145.

[6] Mochon, D. and S. Frederick, *Anchoring in sequential judgments.* Organizational Behavior and Human Decision Processes, 2013. **122**(1): p. 69-79.

[7] Budzier, A. and B. Flyvbjerg, *Overspend? Late? Failure? What the data say about IT project risk in the public sector.* arXiv preprint arXiv:1304.4525, 2013.

[8] Wedell, D.H., S.K. Hicklin, and L.O. Smarandescu, *Contrasting models of assimilation and contrast*, in *Assimilation and contrast in social psychology*, D.A. Stapel and J. Suls, Editors. 2007, Psychology Press: New York. p. 45-74.

[9] Damisch, L., T. Mussweiler, and H. Plessner, *Olympic medals as fruits of comparison? Assimilation and contrast in sequential performance judgments.* Journal of Experimental Psychology: Applied, 2006. **12**(3): p. 166.

[10] Yeates, P., M. Moreau, and K. Eva, *Are examiners' judgments in OSCE-style assessments influenced by contrast effects?* Journal of Academic Medicine, 2015. **90**(7): p. 975-980.

[11] Mussweiler, T., *Focus of comparison as a determinant of assimilation versus contrast in social comparison.* Personality and Social Psychology Bulletin, 2001. **27**(1): p. 38-47.

[12] Schwarz, N. and H. Bless, *Mental construal processes: The inclusion/exclusion model.* Assimilation and contrast in social psychology, 2007: p. 119-141.

[13] Suls, J. and L. Wheeler, *Psychological Magnetism: A Brief History of Assimilation and Contrast in Psychology*, in *Assimilation and contrast in social psychology* Stapel and J. Suls, Editors. 2007, Psychology Press. p. 9-44.

[14] Fechner, G., *Elemente der psychophysik (2 vols). Breitkopf and Härtel.* 1860.

[15] Hellström, Å., *The time-order error and its relatives: Mirrors of cognitive processes in comparing.* Psychological Bulletin, 1985. **97**(1): p. 35-61.

[16] Tversky, A., *Features of similarity.* Psychological Review, 1977. **84**(4): p. 327-352.

[17] Shepperd, M., C. Mair, and M. Jørgensen. *An experimental evaluation of a de-biasing intervention for professional software developers*. In *Annual ACM Symposium on Applied Computing*. 2018. ACM. p. 1510-1517.

[18] Furnham, A. and H.C. Boob, *A literature review of the anchoring effect.* The journal of socio-economics, 2011. **40**: p. 35-42.

[19] Englich, B. and T. Mussweiler, *Sentencing under uncertainty: anchoring effects in the courtroom.* Journal of Applied Social Psychology, 2001. **31**: p. 1535-1551.

[20] Northcraft, G.B. and M.A. Neala, *Experts, amateurs, and real estate: An anchoring-and-adjustment perspective on property pricing decisions.* Organizational Behavior and Human Decision Processes, 1987. **39**: p. 84-97.

[21] König, C.J., *Anchors distort estimates of expected duration.* Psychological Reports, 2005. **96**(2): p. 253-256.

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

[23] Thomas, K.E. and S.J. Handley, *Anchoring in time estimation.* Acta Psychol (Amst), 2008. **127**(1): p. 24-9.

[24] Jørgensen, M. and S. Grimstad, *The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment.* IEEE Transactions on Software Engineering, 2011. **37**(5): p. 695-707.

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

[26] Løhre, E. and M. Jørgensen, *Numerical anchors and their strong effects on software development effort estimates.* Journal of Systems and Software, 2016. **116**: p. 49-56.

[27] Frederick, S.W. and D. Mochon, *A scale distortion theory of anchoring.* Journal of Experimental Psychology: General, 2012. **141**(1): p. 124.

[28] Strack, F. and T. Mussweiler, *Explaining the enigmatic anchoring effect: Mechanisms of selective accessibility.* Journal of Personality and Social Psychology, 1997. **73**(3): p. 437.

[29] Epley, N. and T. Gilovich, *Putting adjustment back in the anchoring and adjustment heuristic: Differential processing of self-generated and experimenter-provided anchors.* Psychological Science, 2001. **12**(5): p. 391-396.

[30] Jørgensen, M. and S. Grimstad, *Software Development Estimation Biases: The Role of Interdependence.* IEEE Transactions on Software Engineering, 2012. **38**(3): p. 677-693.

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

[32] Mussweiler, T., *Comparison processes in social judgment: mechanisms and consequences.* Psychological Review, 2003. **110**(3): p. 472-489.

[33] Thomas, K.E., S.E. Newstead, and S.J. Handley, *Exploring the time prediction process: The effects of task experience and complexity on prediction accuracy.* Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition, 2003. **17**(6): p. 655-673.

[34] Thomas, K.E., S.E. Newstead, and S.J. Handley. *The impact of prior task experience on bias in predictions of duration.* In *Annual Meeting of the Cognitive Science Society.* 2004. p. 1339-1344.

[35] Grimstad, S. and M. Jorgensen, *Preliminary study of sequence effects in judgment-based software development work-effort estimation.* Iet Software, 2009. **3**(5): p. 435-441.

[36] Jørgensen, M., *Relative estimation of software development effort: it matters with what and how you compare.* IEEE Software, 2013. **30**(2): p. 74-79.

[37] Mussweiler, T. and F. Strack, *Numeric judgments under uncertainty: The role of knowledge in anchoring.* Journal of Experimental Social Psychology, 2000. **36**(5): p. 495-518.

[38] Sá, W.C., R.F. West, and K.E. Stanovich, *The domain specificity and generality of belief bias: Searching for a generalizable critical thinking skill.* Journal of educational psychology, 1999. **91**(3): p. 497.

[39] West, R.F., M.E. Toplak, and K.E. Stanovich, *Heuristics and biases as measures of critical thinking: Associations with cognitive ability and thinking dispositions.* Journal of Educational Psychology, 2008. **100**(4): p. 930.

[40] Jørgensen, M. and S. Grimstad, *Avoiding irrelevant and misleading information when estimating development effort.* IEEE Software, 2008. **25**(3): p. 78-83.

[41] Stanovich, K.E. and R.F. West, *On the relative independence of thinking biases and cognitive ability.* Journal of personality and social psychology, 2008. **94**(4): p. 672.

[42] Grinistad, S. and M. Jorgensen, *Inconsistency of expert judgment-based estimates of software development effort.* Journal of Systems and Software, 2007. **80**(11): p. 1770-1777.

[43] Jorgensen, M. and S. Grimstad, *The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment.* IEEE Transactions on Software Engineering, 2010. **37**(5): p. 695-707.

[44] Bergersen, G.R., D.I.K. Sjøberg, and T. Dybå, *Construction and validation of an instrument for measuring programming skill.* IEEE Transactions on Software Engineering, 2014. **40**(12): p. 1163-1184.

[45] van den Berg, R., M. Lindskog, L. Poom, and A. Winman, *Recent Is More: A Negative Time-Order Effect in Nonsymbolic Numerical Judgment.* Journal of

Experimental Psychology-Human Perception and Performance, 2017. **43**(6): p. 1084-1097.

[46] Page, L. and K. Page, *Last shall be first: A field study of biases in sequential performance evaluation on the Idol series.* Journal of Economic Behavior & Organization, 2010. **73**(2): p. 186-198.

[47] Zell, E. and Z. Krizan, *Do people have insight into their abilities? A metasynthesis.* Perspectives on Psychological Science, 2014. **9**(2): p. 111-125.

[48] Nan, N. and D.E. Harter, *Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort.* IEEE Transactions on Software Engineering, 2009. **35**(5): p. 624-637.