The Journal of Systems and Software 117 (2016) 274–281

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays

Magne Jørgensen*

Simula Research Laboratory, Norway

ARTICLE INFO

Article history:
Received 26 November 2015
Revised 16 February 2016
Accepted 17 March 2016
Available online 26 March 2016

Keywords:
Effort estimation
Judgment bias
Effort unit

ABSTRACT

Software development effort estimates are typically expert judgment-based and too low to reflect the actual use of effort. Our goal is to understand how the choice of effort unit affects expert judgement-based effort estimates, and to use this knowledge to increase the realism of effort estimates. We conducted two experiments where the software professionals were randomly instructed to estimate the effort of the same projects in work-hours or in workdays. In both experiment, the software professionals estimating in work-hours had much lower estimates (on average 33%–59% lower) than those estimating in workdays. We argue that the *unitosity* effect—i.e., that we tend to infer information about the quantity from the choice of unit—is the main explanation for the large difference in effort estimates. A practical implication of the unit effect is that, in contexts where there is a tendency toward effort under estimation, the instruction to estimate in higher granularity effort units, such as workdays instead of work-hours, is likely to lead to more realistic effort estimates.

1. Introduction

Software development projects have a reputation of being late and with large cost overruns. Surveys suggest that software projects on average overrun their budgets by 20–30% (Halkjelsvik and Jørgensen, 2012) and their expected duration by about 20% (Moløkken and Jørgensen, 2003). The proportion of software projects with very high budget overruns and time delays is substantial. A recent, large-scale review of cost and time overruns in public IT reports that the 18% worst performing projects had an average budget overrun of 130% and an average time overrun of 41% (Budzier and Flyvbjerg, 2012). Effort underestimates and consequent unrealistic plans and budgets may lead to severe problems for both the software providers and their clients (Flyvbjerg and Budzier, 2011). Similar tendencies toward time and budget overruns are documented for several other types of engineering projects (Flyvbjerg et al., 2003). In spite of much research on the reasons for effort estimation overoptimism (van Genuchten, 1991; Moses and Farrow, 2003; Jørgensen and Moløkken-Østvold, 2004; Jørgensen et al., 2004), the challenge of realistically estimated software projects remains.

While the use of formal effort estimation models is common in many engineering disciplines, the prevalent effort estimation method in software development projects is judgment-based (Jørgensen and Shepperd, 2007). One reason for this is that expert judgment-based effort estimates of software projects generally are at least as accurate as model-based effort estimates (Jørgensen, 2007). However, there is also strong evidence that judgment-based effort estimates can easily be biased toward overoptimism (Halkjelsvik and Jørgensen, 2012) and a need to look for ways to improve such estimates.

This paper addresses how the choice of effort unit used in software project estimation may affect the effort estimates, including the tendency toward over-optimism. As far as we know, the effect of the choice of the effort unit has not yet been studied or proposed as a factor contributing to the tendency toward overoptimistic effort estimates. Recent findings in other domains suggest however the presence of strong unit-related effects on decisions. As an illustration, when experienced judges were asked to suggest a sentence in a realistic, experimental trial in months, they decided on an average sentence of 5.5 years (66.4 months), while when asked to do the same in years, they decided on an average sentence of 9.7 years (Rachlinski et al., 2015). Maybe for the same reason, i.e., the effect of the unit on sentence decisions, in 1991 Finnish courts were instructed to set the shorter sentences in days instead of months to shorten the sentence lengths. After implementing this change in practice the sentence length actually decreased (Lappi-Seppälä et al., 2001).

Previous results on unit-related effects focus on how decisions, such as donations or sentence lengths, are influenced by the

* Tel.: +47 924 333 55.
E-mail address: magnej@simula.no

http://dx.doi.org/10.1016/j.jss.2016.03.048
0164-1212/© 2016 Elsevier Inc. All rights reserved.

perception on how much, how large or how long something is in different units, e.g., whether dieting for 1 year feels longer than dieting for 12 months. The unit effects in the decision-oriented situation are, at least potentially, quite different from the prediction situations we examine in this paper. An estimate of the effort required to complete a task involves the *generation* of numbers (the candidate effort estimates) when presenting the unit to be used, while the decision-oriented situations are based on *presentation* of both numbers and their units. Not only are there potentially differences between the unit effects, i.e., how much, large or long something is, from self-generated and presented numbers, there are, as we will discuss later in this paper, a difference in the direction of potentially important effects in the two situations. The lack of, as far as we are aware of, previous results on unit effects in prediction situations, led us to believe that our studies had the potential to contribute with novel results.

The effort unit used for software development effort estimation is sometimes work-hours and sometimes other time units, such as workdays. [1] The question addressed in this paper is whether the choice of effort unit affects the effort estimates and to what extent this effect is of practical importance. If a client or manager asks a software professional about the effort needed to complete a software task or project, would it matter whether the request is formulated, for example, as “How many work-hours do you think it will take?” or as “How many workdays do you think it will take?”

Rationally speaking, an effort estimate should not be much affected by the unit chosen. An estimate in one unit, e.g., workdays, can be converted into an estimate in another unit, e.g., work-hours, representing the same amount of effort. In spite of this, we have experienced that many software professionals believe that effort estimation is not unit-neutral. We conducted an online survey with 77 software professionals from Norway, with experience in effort estimation. The software professionals were asked the following question: “How do you believe the effort unit (work-hours or workdays) typically affects, if at all, the estimated effort?” The survey revealed a tendency toward believing that effort estimation in work-hours leads to lower rather than higher estimates: 56% believe in lower estimates, while 32% believed in higher estimates when calculated in work-hours and the remaining (12%) believed in no difference. Twenty-six percent of them thought that the effect would lead to an increase or decrease of the effort estimates of more than 20%.

Based on the results in Rachlinski et al. (2015) suggesting that a lower granularity time unit gives lower time estimates, and the (weak) tendency of the software professionals to believe in a decrease in effort estimates when using work-hours instead of workdays, we hypothesized that the unit effects found in decision-oriented situations would be analogous to those found in prediction-oriented situations:

Hypothesis: Estimating effort in work-hours leads to lower estimates than corresponding estimates in workdays.

Notice that there are theories and mechanisms that potentially would predict an effect in the opposite direction as the one we hypothesize. This includes:

- *The construal level theory*. This theory aims at explaining the relation between the psychological distance to the subject of our thinking and the abstraction level of the thinking. The construal level theory predicts that higher psychological distance leads to higher construal level, i.e., higher abstraction level and more goal-oriented thinking (Trope and Liberman, 2010). The relevance of distance in time for abstraction and goal-orientedness of thinking has been documented in for example (Liberman and Trope, 1998; Maglio and Trope, 2011). The relation between construal level and unit-effects is reported in (Monga and Bagchi, 2012). The experiments reported in (Kanten, 2011; Yan et al., 2014) find that an increase in construal level, i.e., higher granularity time-related units, increased the estimates. The experiments in (Siddiqui et al., 2014), on the other hand, report that the estimates increased with increase in construal level for simple tasks only. The estimates decreased for more complex tasks. A higher granularity time unit, such as workdays, may consequently lead to lower effort estimates for complex tasks.
- *The fluency theory*. We may process mental work, such as estimating required work-hours of software development activities, with ease or we may find it difficult and inefficient. The subjective experience of ease with which we complete mental processes is what we mean by cognitive fluency (Oppenheimer, 2008). Our judgments may be affected by the fluency of the mental process producing the judgment. We may for example use the fluency of the mental process trying to understand a requirement specification as an indicator of how difficult the software work will be. In Song and Schwarz (2008), for example, it is reported that easy-to-read instructions, high font readability, produced lower time estimates than hard-to-read instructions, low font readability, for the same task. In Ülkümen et al. (2008) the default (most fluent) unit for judging product characteristics led to the most favorable evaluations and explain this as caused by processing fluency. Applied on or examination of the effect of different effort estimation units, we may experience that if the effort unit feels less natural to use, such as estimating large tasks in work-hours rather than workdays, the estimate may increase.

The reason for keeping the hypothesis that estimating in workdays leads to higher estimates than estimating in work-hours is based on that the potential effects in the opposite directions are typically small and that the two main effects typically used to explain the unit effects, i.e., numerosity (Ülkümen and Thomas, 2013) and unitosity (Ülkümen et al., 2008; Monga and Bagchi, 2012), are typically much larger. We discuss numerosity and unitosity and how they may explain the observed unit-related effects in Section 4.

The rest of this paper is organized as follows: Section 2 describes the design and results of our first experiment on the effect of the choice of time unit on the effort estimates. Section 3 replicates and extends the first experiment using a different population, a different requirement specification, and a different estimation method. The second experiment also includes a comparison of the estimates with the actual effort seven other providers used to complete the software development project. Section 4 tries to explain the observed differences using established theories about human judgment. Section 5 describes practical implications of the results, draws conclusions, and outlines further research.

2. Experiment 1: effort unit effects in Work Breakdown Structure-based estimation

2.1. Experiment design

The experiment was designed and executed as follows:

[1] A re-analysis of the data set of 230 small and medium-large projects published in (Jørgensen and Grimstad 2011) indicates that 89% of the activity estimates were in work-hours and 11% in workdays. The larger the projects, the more likely they were to be estimated using workdays.

- Development of two requirement specifications, one specifying a simple web service (Project A) and the other specifying an application visualizing connections between countries on a world map (Project B). Project A was considered to be quite a small project, while Project B was a more comprehensive and complex project.[2] A difference in project sizes was introduced to examine to what extent an effort unit effect, if any, depended on the size of the project.
- Recruitment of software professionals from two offshoring companies, one in Poland and one in Romania. The work was paid work using the companies' ordinary hourly rates. The software professionals had at least one year of experience as software developers or project managers. All of the software professionals had previous experience in estimating software development work and with the type of software work, including the technologies and programming languages, to be estimated.
- Collection of information about the role and experience of each of the participants. We collected information about their current role, the length of their experience as software professionals, the amount of estimation experience, and their self-assessed skill as software developers.
- Estimation of the effort of Projects A and B, individually, by each of the participants. The participants were instructed to use a Work Breakdown Structure-based estimation approach (Tausworthe, 1980). The participants were also instructed to describe technical and design-related decisions, identify the required activities, and estimate the most likely effort, the minimum effort, and the maximum effort of each activity. The estimation of the minimum and maximum effort enabled us to see to what extent the effort unit had an impact on the width of the effort prediction interval,[3] that is, how uncertain the participants thought their effort estimates were. The participants were instructed to assume that they would do all the work themselves. Each participant was randomly allocated either to a group estimating in workdays (Group Workdays) or to a group estimating in work-hours (Group Work-hours). The sequence of the estimation of the two projects, i.e., whether first estimating Project A and then Project B or first estimating Project B and then Project A, was also randomized. After the completion of the estimation of both projects, the participants estimating in workdays were asked to provide a conversion factor between workdays and work-hours, i.e., how many work-hours they usually would consider as constituting one workday.

To assess the number of participants required for our first experiment, we conducted a statistical power analysis (Cohen, 1992). We assumed a medium-large effect size (Cohen's d of 0.6), motivated by what we would consider to be a substantial unit effect and the effect size of the relatively similar time unit experiment for judges reported in (Rachlinski et al., 2015), a Type I error (significance level) of 0.05, a Type II error of 0.3, and a one-sided t-test of difference in mean values, with unequal variance. With these assumptions we would need 27 participants in each group, i.e., 54 participants in total. Notice that a Type II error of 0.3 gives a statistical power of 70%, which means that it is at least 70% likely that we find a statistically significant ($p < 0.05$) difference if there is a true difference of at least the assumed effect size. This gives a balance between Type I errors (5% likely to falsely claim that there is a difference) and Type II errors (30% or less likely to find no difference if there is a difference with at least the assumed effect size).

2.2. Results

2.2.1. Participant characteristics

We recruited a total of 74 software professionals, i.e., more than the required number of participants suggested by the power analysis. Eighty-nine percent of the software professionals stated that their main role was as software developers. The remaining 11% indicated that their role was as project managers or team leaders. Table 1 shows other information about the participants, suggesting that there were only small differences between the two estimation groups. Those estimating in workdays (Group Workdays) spent, on average, slightly more time on the estimation tasks than those estimating in work-hours (Group Work-hours). This corresponds with an observation that those estimating in workdays included an average of 38% more activities in their Work Breakdown Structure for Project B and about the same number of activities for Project A; i.e., they were a bit more thorough in their estimates. The observed, statistically non-significant difference in time spent and number of identified activities could be caused by random variation, but may potentially be a consequence of the effort unit that the participants were instructed to use.

2.2.2. Difference in effort estimates

Tables 2 and 3 display the characteristics of the effort estimates of the two groups for Project A (the smaller project) and Project B (the larger project). We used the conversion factor between workdays and work-hours provided by the software professional in Group Workdays to calculate the estimated number of work-hours. The mean conversion factor was 7.1 work-hours per workday, varying from 6 to 8 work-hours per workday. Notice that the conversion factor question asked how the participants would normally go from the number of estimated workdays to the number of estimated work-hours, not how many productive work-hours they thought they would have during a normal calendar workday.

The data in Table 2 supports our hypothesis that people decrease their estimates when estimating in work-hours instead of workdays. There is only a 1% ($p = 0.01$) likelihood of observing the collected data (or more extreme data) given that estimating in work-hours would lead to higher or equal estimates. A more outlier robust statistical test, i.e., the Kruskal–Wallis test of differences in median values (corrected for ties), gives $p < 0.001$, based on median estimates of 50 work-hours for those in Group Workdays and 21 work-hours for those in Group Work-hours. The effect size, measured as Cohen's d, is medium ($d = 0.54$). We found no large or statistically significant differences in the relative width (RWidth) of the two groups' respective effort prediction intervals. The distributions of effort estimates were strongly right-skewed, but assessed to be sufficiently close to a normal distribution to allow the robust t-test of difference in mean values. The similarity of results with the non-parametric Kruskal–Wallis test supports this.

Table 3 shows that our hypothesis of lower estimates when estimating in work-hours was supported by the collected estimation data for Project B, as well. It is only 0.3% ($p = 0.003$) likely to observe the collected data or more extreme data given that estimating in workdays leads to lower or same estimates. The Kruskal–Wallis test results in $p = 0.009$, based on median estimates of 276 work-hours for those in Group Workdays and 198 work-hours for those in Group Work-hours. The effect size, measured as Cohen's d, is medium ($d = 0.64$). As with Project A, there were no large or statistically significant differences in the relative width (RWidth) of the effort prediction interval.

In a previous study (Grimstad and Jørgensen, 2009), we found an effect of the sequence of estimating software projects. Estimating a medium-large project just after estimating a small project gave lower estimates than when estimating the same medium-large project after a larger project. We proposed an explanation

[2] To receive the requirement specifications, please contact the author of this paper.

[3] To measure the (relative) width of the effort prediction intervals we use the measure RWidth = (Maximum effort – Minimum effort) / Most likely effort.

Table 1
Information about the participants.

Group	n	Time spent (mean)	Developer experience (mean)	Developer skill[a] (mean)	Estimation experience[b] (mean)
Workdays	36	139 min	6.2 years	2.2	3.6
Work-hours	38	117 min	6.2 years	2.1	3.4
Total	74	128 min	6.2 years	2.1	3.5

[a] We used the scale: "very good" (1) – "good" (2) – "average" (3) – "low" (4) – "very low" (5). 97% of the professionals assessed their programming skill to be average (for their company) or better. None gave the value "very low".

[b] Scale used: "more than 50 projects estimated" (1) – "20–50 projects estimated" (2) – "5–20 projects estimated" (3) – "1–4 projects estimated" (4) – "no projects estimated" (5). In total 6 software professionals (5 in the workday's estimation group and 1 in the work-hour group) indicated that they had never estimated any project. Upon closer examination we found that they were still qualified since they had extensive experience with estimating tasks and sub-projects within projects.

Table 2
Estimates of Project A.

	Most likely effort (mean)	RWidth (mean)[a]
Estimates in workdays (converted)	88 work-hours (std. dev. 94)	0.88 (std. dev. 0.4)
Estimates in work-hours	45 work-hours (std. dev. 56)	0.82 (std. dev. 0.3)
Difference between Groups	43 work-hours (49% decrease)	0.06 (7% decrease)
t-test of difference (p-value)	0.01[b]	0.24[c]

[a] RWidth is measured as: (Maximum effort – Minimum effort)/Most likely effort.

[b] One-sided t-test, reflecting our one-sided hypothesis, assuming unequal variance.

[c] Two-sided t-test, reflecting no prior hypothesis about the direction of an effect, assuming unequal variance.

Table 3
Estimates of Project B.

	Most likely effort (mean)	RWidth (mean)
Estimates in workdays (converted)	335 work-hours (std. dev. 193)	0.88 (std. dev. 0.3)
Estimates in work-hours	224 work-hours (std. dev. 133)	0.82 (std. dev. 0.3)
Difference between Groups	111 work-hours (33% decrease)	0.06 (7% decrease)
t-test of difference (p-value)	0.003[a]	0.24[b]

[a] One-sided t-test, reflecting the one-sided hypothesis, assuming unequal variance.

[b] Two-sided t-test, reflecting no prior hypothesis about the direction an effect, assuming unequal variance.

based on "the assimilation-contrast theory" (Sherif and Hovland, 1961) in that paper (Grimstad and Jørgensen, 2009). A similar effort estimation sequence effect was present in the current study. When the smaller project (Project A) was estimated first, the mean estimate of the larger project (Project B) was 32 work-hours lower than when starting with the larger project. Similarly, when the larger project (Project B) was estimated first, the mean estimate of the smaller project (Project A) was 45 work-hours higher than when starting with the smaller project. The difference was statistically significant for the estimates of Project A ($p = 0.01$), but not for project B ($p = 0.20$). The sequence effect was similar for those estimating in work-hours and those estimating in workdays, i.e., the effort unit effect seems to be independent of the sequence.

We examined the extent to which the effort unit effect depended on the participants' main role (project leader or software developer), the amount of estimation experience (in number of times they had estimated a project), and their self-assessed skill as software developers. No statistically significant connections were found. If anything, the effort unit effect slightly increased with longer experience as a software professional, more estimation experience and higher self-assessed skill. This suggests that the unit effect we observed was not mainly an effect on less experience or skill.

The participants in Experiment 1 applied an estimation approach based on a Work Breakdown Structure, i.e., a method that involves the identification and estimation of activities or deliverables needed to complete a software project. Experiment 2 assesses the robustness of the unit effect found in Experiment 1 by using a different estimation approach, i.e., analogy-based estimation (Hihn and Habib-Agahi, 1991), a different requirement specification, and a different population of software professionals.

3. Experiment 2: effort unit effects in analogy-based estimation

3.1. Experiment design

Experiment 2 was designed and executed as follows:

- Development of a requirement specification. We applied a slightly reduced version of a requirement specification included in a previous experiment. We knew the actual effort spent by seven different offshoring companies completing the specified work (Jørgensen, 2016).
- Recruitment of 55 software professionals from a large multinational software company. These software professionals were mainly experienced project managers, but also included general managers and other roles with estimation experience. We stopped the recruitment after reaching 55 participants, assuming that the statistical power analysis from Experiment 1 was still valid and that this number would be sufficient. The participants were requested to estimate the software development effort of a typical development team in their own company for software, as specified by us. The requirement specification described a web-application for storing and retrieving data about scientific articles.
  - ○ Using the experiment/survey software Qualtrics (www.qualtrics.com) we instructed the participants to follow an

analogy-based estimation process. The instructed process was as follows:

- "Identify analogies, i.e., tasks or projects with deliverables similar in size and complexity to the specified application, for which you know the effort used to complete the work."
- "Use the identified analogies, or if none found, your expert judgment, to estimate the effort a typical developer at your company would spend to complete the specified application." The participants were, as in the previous experiment, randomly instructed to estimate in work-hours (Group Work-hours) or in workdays (Group Workdays). We did not ask for effort prediction intervals in this experiment.
- "Indicate the number of analogies identified and the process used to derive the estimated effort from the analogies."
- "Provide the conversion factor between workdays and work-hours" (only relevant for those in Group Workdays).

3.2. Results

The number of analogies identified by the participants was distributed as follows: 4 participants found more than 5 analogies, 18 found 2–5 analogies, 18 found 1–2 analogies, while 16 found no analogies (and used purely their "expert judgment" to give an effort estimate of the project). There were no noticeable differences between the two effort unit groups in terms of the number of analogies identified.

The mean conversion factor between workdays and work-hours was 7.3, ranging from 4 to 8 work-hours. Eighty-seven percent of the participants used a conversion factor between 7 and 8 work-hours.

An examination of the described analogy-based estimation processes revealed that seven of the participants instructed to estimate in work-hours reported that they actually had estimated in workdays (3), work-weeks (1) or man-months (3) and converted that figure to work-hours. This may be due to the fact that a higher granularity effort unit is felt to be more natural for an estimation process based on recalling the total effort of previously completed projects. That is, the total effort may be more likely to be recalled in workdays, workweeks or man-months. The mean estimate of those seven participants was 199 work-hours, which was close to the mean estimate of those in the workdays group (mean of 178 work-hours), and much higher than those in the remaining work-hours group (mean of 72 work-hours). We chose to remove these seven participants from the analysis. Inclusion of them in Group Work-hours would not be valid, since they did not estimate in work-hours. To change their group allocation to Group Workdays would not be valid either since we did not know whether their effort estimates were higher because they estimated in a higher granularity unit, or whether they estimated in a higher granularity unit because they thought the project was larger.

The resulting mean estimates for each group, after removing the seven participants, are displayed in Table 4. The difference in mean estimates is statistically significant and with an effect size (Cohen's *d* of 0.52) similar to those in Experiment 1, which had Cohen's *d* of 0.54 and 0.64 respectively. Even if we include the seven participants who did not follow the instruction to estimate in work-hours, the difference is still there (now with Cohen's *d* of 0.35), although not statistically significant (a *t*-test yields $p = 0.09$ and a Kruskal–Wallis test yields $p = 0.05$). The distributions of effort estimates were strongly right-skewed, but assessed to be sufficiently close to a normal distribution to allow the robust *t*-test of difference in mean values. The similarity of results with the non-parametric Kruskal–Wallis test (corrected for ties) supports the correctness of this assumption.

Table 4
Estimates of the Web-project.

	Most likely effort (mean)
Estimates in workdays, $n = 29$	177 work-hours (std. dev. 254)
Estimates in work-hours, $n = 19$	72 work-hours (std. dev. 57)
Difference between Groups	105 work-hours (59% decrease)
t-test of difference (*p*-value)	0.02[a]

The software application estimated by the participants in our experiment had already been developed by seven different companies as part of another experiment (Jørgensen, 2016). The main difference between the specifications given to these seven companies and those used in the experiment was the removal of an import feature to make the specifications easier to read. The import feature typically took about 10% of the total work effort for the seven companies. The mean actual effort used by the seven companies to complete the project was 257 work-hours. Removing 10% of that effort gives a mean actual effort of around 240 work-hours, i.e., much higher than the mean estimate of those in Group work-hours and also higher that the estimate of those in Group workdays. Only one company managed to complete the work with the amount of effort estimated by those in Group Work-hours. This suggests that those in Group Work-hours tended to be strongly overoptimistic. Consequently, the use of workdays as the effort estimation unit for this context would most likely have led to more accurate, though still perhaps overoptimistic effort estimates.

[a] One-sided *t*-test, reflecting the one-sided hypothesis, assuming unequal variance.

4. Explaining the effort unit effect

Our experiments suggest that the choice of unit affected the effort estimates. The effect sizes were all medium-large, with Cohen's *d* values in the range of 0.52–0.64. The impact was present within three different estimation tasks, two populations of software professionals, and two different estimation approaches. This suggests a rather robust and important effect.

Research on human judgment has identified several mechanisms and theories that connect a difference in unit to a difference in judgment. This includes the construal level theory (Trope and Liberman, 2010; Kanten, 2011; Maglio and Trope, 2011; Siddiqui et al., 2014; Yan et al., 2014), the fluency theory (Oppenheimer, 2008; Song and Schwarz, 2008; Ülkümen et al., 2008), the central tendency effect (Hollingworth, 1910; Frederick et al., 2011; Tamrakar and Jørgensen, 2012), and effects of rounding numbers (Huttenlocher et al., 1990; Cannon and Cipriani, 2006). All these theories may explain smaller effort unit effects, see for example our discussion on the construal level and fluency theory in Section 1, but hardly the large effect sizes observed in our experiments. Only two mechanisms were found relevant to explain the effect sizes observed in our experiments: the numerosity and the unitosity effects.

4.1. The numerosity effect

The numerosity effect implies that we are frequently affected not only by the real value of something, but also by its nominal value. An example of the numerosity effect is the observed 11% increase in donations given by church visitors between 2002 and 2003, when Italy went from lire (high nominal values) to euros (low nominal values), despite an income growth of only 3% (Cannon and Cipriani, 2006). According to the numerosity effect, church visitors felt that the low nominal number of euros was less than the high number of lire they normally donated, and tended to give more in euros than in lire. Notice that this natural experiment illustrates that numerosity effects are not dependent on that both the number and the unit are presented together. It is sufficient that the unit is presented and that the numbers are self-generated, just as in our experiments.

Ülkümen and Thomas (2013) report experiments relating numerosity to the feeling of duration; the authors found that higher numbers resulted in the feeling of longer durations. For example, most participants felt that 365 days were longer than 12 months. A practical implication of the numerosity effect was that people were more willing to start a diet when framed as a 12-month, rather than a 365-day plan. The numerosity effect increased with greater personal relevance of dieting.

The numerosity effect may imply that people think an effort value in a lower granularity unit, such as work-hours instead of workdays, can contain more work than the corresponding effort value in a higher granularity effort unit, though rationally speaking it involves the same level of work effort. For example, if we assume that one day's work is equal to about 7 work-hours, the numerosity effect would predict that we think we could do more in 35 work-hours (which has a higher nominal value) than 5 workdays (which has a lower nominal value).

4.2. The unitosity effect

The unit itself may also affect the estimate (Ülkümen et al., 2008; Monga and Bagchi, 2012), e.g., due to conversational norms (Grice, 1975) and/or through an anchoring effect (Mussweiler and Strack, 2001).

The effect of conversational norms is related to the fact that we usually request estimates of smaller amounts using lower granularity units and higher amounts using higher granularity units. For example, we tend to use the requested effort or time unit as an indication of whether a project is expected to be small or large. In such case, the request "How many man-years will this work require?" shows different expectations than those implied by "How many work-hours will this work require?" The effort expectation of the person requesting an estimate has been documented to have an effect on the estimates, even when the estimators know that the expectation is from an incompetent, uninformed source (Jørgensen and Sjøberg, 2004; Løhre and Jørgensen, 2016).

The anchoring effect, with the "selective accessibility" mechanisms proposed in (Mussweiler and Strack, 2001), is based on knowledge activation. It is typically related to numerical anchors, but can also include non-numerical anchors. Estimating in a low granularity effort unit may, according to this mechanism, make the estimator more likely to activate and apply knowledge about larger projects and more complex tasks (Strack and Mussweiler, 1997). An example of the effect of textual anchors is documented in one of our earlier studies. Our request to estimate "a minor modification" of a software product resulted in a 50% lower effort estimate than the request to estimate "a new functionality" for exactly the same software product update (Jørgensen and Grimstad, 2008).

4.3. Explanatory strength of numerosity and unitosity

An effort estimate contains both a numerical value and an effort unit. This means that it may be hard to separate the relative influence of the numerical value (numerosity) and its unit (unitosity) on the effort estimates. To our knowledge, the only paper trying to study the relative importance of the two effects is by (Monga and Bagchi, 2012). They found that people could be influenced to focus on the numerical value, in which case the numerosity effect dominated, but also to focus on the unit, in which case the unitosity effect dominated. In the effort estimation contexts of our present study, as opposed to the contexts studied in (Monga and Bagchi, 2012), numerosity and unitosity affect the estimates in the same direction and are more difficult to separate. It would be dangerous to assume similar effect sizes given the different domains and judgment situations. A potentially important difference is between situations where the number and the unit is presented, as in (Monga and Bagchi, 2012), and in situations with self-generated numbers, where only the unit is presented, as in our experiments. From anchoring studies we know that there may be important differences in the effect from presented and self-generated numbers (Epley and Gilovich, 2001).

There are reasons to believe that the effect from the unit itself (unitosity) is a more likely explanation of our findings than an effect from the numerosity:

- Only the unit, not the number, is part of the estimation request. According to Monga and Bagchi (2012), when the focus is on the unit, the unitosity effect will outweight the numerosity effect. Notice, however, that this argumentation assumes that this focus on the unit is not outweighted by a potentially stronger effect of self-generated instead of presented numbers.
- The substantial effect sizes in our studies are larger than those typically reported in numerosity effect studies, such as (Cannon and Cipriani, 2006), but similar to those found in studies relating textual anchors to effort and time judgments, e.g., (Jørgensen and Grimstad, 2008; Rachlinski et al., 2015).

Based on the limited and, admittedly, not very strong argumentation we think that unitosity is the main candidate effect for explaining the large unit effects we observed in our effort estimation context. This question should, however, be subject to further studies.

5. Limitations

In this paper we test whether there is a unit effect on effort estimates through controlled experiments and not through analysis of data from completed projects. There were two main reasons for not using observational data for our purpose. Firstly, it would have been difficult to know to what extent a higher effort estimate was the consequence of a larger, more complex project or of the chosen effort unit. Secondly, even if we had found comparable tasks estimated using different effort units, we would have had difficulties determining whether a project was estimated as requiring more effort due to using a higher granularity effort unit, or whether it used a higher granularity unit because it was believed to be large. In other words, the direction of the causality would have been difficult to establish. The software professionals' varying opinions about the effect of the unit on the effort estimates may to some extent be a result of the problem of choosing between alternative explanations for effort estimation differences in contexts with observational data.

Experimental contexts—even when including software professionals, normal hourly payment, and realistic requirement specifications as we do in our experiments—have limitations when it comes to the external validity. In our case, we believe that the main limitation is related to the artificial context of the experiment. The software professionals knew that their estimates would not be used for the purpose of bidding, planning or something else, but were part of an academic study on effort estimation. It is possible that they would have estimated more thoroughly, spent more time analysing the task, collected more information, and been less affected by the effort unit, if the accuracy of the estimate had mattered more. A randomized controlled experiment in a field setting would consequently have increased the external validity. In a previous study, including an experiment on similar anchoring effects (which we have argued as being the underlying mechanism for the observed unit effect) we found that field settings gave results in the same direction as the laboratory-based experiments. The effect sizes were, however, 30–50% lower (Jørgensen and Grimstad, 2011). We should therefore not expect the impact of the effort unit to be as large in field settings as observed in our experiments. What we have shown is, however, that there is an effort unit effect. Even with a 30–50% decrease in effect size, the effort unit effect is of relevance for designing good estimation processes.

We should be careful when generalizing the results found in this paper to apply to other effort or time estimation contexts. The software projects we presented are typical and those who estimated have extensive, relevant estimation experience. There may nevertheless be context effects of which we are currently unaware. In particular, there may be effects related to whether the person asking for an estimate using a particular effort unit is considered to be a competent source or not. According to Gricean norms of conversation, the effect should be lower when from an incompetent rather than a competent source. In our experiments, those who estimated had reasons to believe that the request was from a reasonably competent source, which consequently may have enhanced the effect compared to when coming from a source with little or no relevant competence. However, in earlier studies we have found that there is a substantial remaining effect in similar effort estimation anchoring contexts even when explicitly explaining that the source had no relevant competence (Jørgensen and Sjøberg, 2001; Løhre and Jørgensen, 2016). We would consequently expect that the effect would be present, perhaps lower, even in contexts with low competence sources.

Our study is about expert judgment-based effort estimates. The participants did not have access to historical data, estimation models or used estimation learning frameworks such as the Personal Software Process (Humphrey, 1996; Kamatar and Hayes, 2000; Prechelt and Unger, 2000). We expect the unit effect still to be present with more explicit use of historical data and the use of formal models and frameworks, because the selection of relevant data and the use of formal effort estimation models will need expert judgement. The size of the unit effect in effort estimation situations may, however, be substantially reduced.

Those estimating the effort in workdays were asked to provide the number of work-hours per day they would use when converting from work-hours to workdays. We tried to formulate this to make it clear that we did not ask about how many productive hours they had each day, but rather how they went from work-hours to workdays in their effort estimates. It is nevertheless, possible that a few, such as those calculating only 4 work-hours per workday misunderstood this and included less time waste in their estimates than those who estimated in workdays. Fortunately, the observed unit effect is large enough to exclude that a difference in how they interpreted an effort estimate is not likely to explain the all of the observed differences in estimates.

The above potential limitation points at an interesting topic for further studies: Will those estimating in work-hours think about software development work as including less waste of time (interruptions, misunderstandings, low productivity situations etc.) than those estimating in workdays? A difference in thinking could, for example, be present if those estimating in work-hours were splitting tasks into smaller sub-tasks than those estimating in workdays and not counting the waste in-between the sub-tasks as part of their estimate. Formulated differently, are those estimating in workdays more likely to think about time waste as an integrated part of what they do?

6. Conclusions and implications

Software professionals tend to give substantially lower effort estimates for the same projects when estimating in work-hours rather than workdays. We argue that the use of the effort unit is understood as an indicator of the expected effort of the project, e.g., through Gricean norms of conversation and/or anchoring effects, and that this is the underlying mechanism explaining the effort unit effects. Higher granularity of the effort unit may mean that people start searching for experience from larger tasks and activities than with lower granularity effort units, and that they use this recently activated experience when estimating the required effort.

Our findings imply that the choice of effort unit matters for the accuracy of effort estimation and, consequently, for the successful planning and execution of projects. The choice of effort unit may be particularly important in estimation contexts where there is a tendency toward underestimating effort, such as when the experience is low, the uncertainty is high, and/or the competition among vendors is strong (Lederer and Prasad, 1995; Chan and Kumaraswamy, 1997; Bubshait, 2003; Eden et al., 2005; Jørgensen, 2006). In such contexts, the risk of cost overrun would increase with a low granularity effort unit, such as work-hours, and the estimation task would benefit from using higher granularity effort units, such as workdays.

References

Bubshait, A.A., 2003. Incentive/disincentive contracts and its effects on industrial projects. Int. J. Project Manage. 21 (1), 63–70.

Budzier, A., Flyvbjerg, B., 2012. Overspend? Late? Failure? What the data say about IT project risk in the public sector. Commonwealth Governance Handbook, 13, pp. 145–157.

Cannon, E.S., Cipriani, G.P., 2006. Euro-illusion: a natural experiment. J. Money, Credit, Banking 38 (5), 1391–1403.

Chan, D.W.M., Kumaraswamy, M.M., 1997. A comparative study of causes of time overruns in Hong Kong construction projects. Int. J. Project Manage. 15 (1), 55–63.

Cohen, J., 1992. Statistical power analysis. Curr. Dir. Psychol. Sci. 1 (3), 98–101.

Eden, C., Williams, T., Ackermann, F., 2005. Analysing project cost overruns: comparing the “measured mile” analysis and system dynamics modelling. Int. J. Project Manage. 23 (2), 135–139.

Epley, N., Gilovich, T., 2001. Putting adjustment back in the anchoring and adjustment heuristic: differential processing of self-generated and experimenter-provided anchors. Psychol. Sci. 12 (5), 391–396.

Flyvbjerg, B., Budzier, A., 2011. Why your IT project may be riskier than you think. Harvard Bus. Rev. 89 (9), 601–603.

Flyvbjerg, B., Skamris Holm, M.K., Buhl, S.L., 2003. How common and how large are cost overruns in transport infrastructure projects? Transp. Rev. 23 (1), 71–88.

Frederick, S.W., Meyer, A.B., Mochon, D., 2011. Characterizing perceptions of energy consumption. Proc. Natl. Acad. Sci. USA 108 (8), E23.

Grice, H., 1975. Logic and conversation. In: Cole, P., Morgan, J. (Eds.), Syntax and Semantics Volume 3: Speech Acts. Academic Press, New York.

Grimstad, S., Jørgensen, M., 2009. Preliminary study of sequence effects in judgment-based software development work-effort estimation. IET Software 3 (5), 435–441.

Halkjelsvik, T., Jørgensen, M., 2012. From origami to software development: a review of studies on judgment-based predictions of performance time. Psychol. Bull. 138 (2), 238–271.

Hihn, J., Habib-Agahi, H., 1991. Cost estimation of software intensive projects: a survey of current practices. In: International Conference on Software Engineering, Austin, TX, USA. IEEE Comput. Soc. Press, Los Alamitos, CA, USA, pp. 276–287.

Hollingworth, H.L., 1910. The central tendency of judgment. J. Philos. Psychol. Sci. Method 7 (17), 461–469.

Humphrey, W.S., 1996. The PSP and personal project estimating. Am. Programm. 9 (6), 2–15.

Huttenlocher, J., Hedges, L.V., Bradburn, N.M., 1990. Reports of elapsed time: bounding and rounding processes in estimation. J. Exp. Psychol. Learn. Memory Cognit. 16 (2), 196.

Jørgensen, M., 2006. The effects of the format of software project bidding processes. Int. J. Project Manage. 24 (6), 522–528.

Jørgensen, M., 2007. Forecasting of software development work effort: evidence on expert judgement and formal models. Int. J. Forecasting 23 (3), 449–462.

Jørgensen, M., 2016. Better selection of software providers through trialsourcing. IEEE Software (in press).

Jørgensen, M., Grimstad, S., 2008. Avoiding irrelevant and misleading information when estimating development effort. IEEE Software 25 (3), 78–83.

Jørgensen, M., Grimstad, S., 2011. The impact of irrelevant and misleading information on software development effort estimates: a randomized controlled field experiment. IEEE Trans. Software Eng. 37 (5), 695–707.

Jørgensen, M., Moløkken-Østvold, K., 2004. Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method. IEEE Trans. Software Eng. 30 (12), 993–1007.

Jørgensen, M., Shepperd, M., 2007. A systematic review of software development cost estimation studies. IEEE Trans. Software Eng. 33 (1), 33–53.

Jørgensen, M., Sjøberg, D.I.K., 2001. Impact of effort estimates on software project work. Inf. Software Technol. 43 (15), 939–948.

Jørgensen, M., Sjøberg, D.I.K., 2004. The impact of customer expectation on software development effort estimates. Int. J. Project Manage. 22, 317–325.

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

Kamatar, J., Hayes, W., 2000. An experience report on the personal software process. IEEE Software 17 (6), 85–89.

Kanten, A.B., 2011. The effect of construal level on predictions of task duration. J. Exp. Soc. Psychol. 47 (6), 1037–1047.

Lappi-Seppälä, T., Tonry, M., Frase, R., 2001. Sentencing and Punishment in Finland: the Decline of the Repressive Ideal. Oxford University Press, New York, NY.

Lederer, A.L., Prasad, J., 1995. Causes of inaccurate software development cost estimates. J. Syst. Software 31 (2), 125–134.

Liberman, N., Trope, Y., 1998. The role of feasibility and desirability considerations in near and distant future decisions: a test of temporal construal theory. J. Pers. Soc. Psychol. 75 (1), 5.

Løhre, E., Jørgensen, M., 2016. Numerical anchors and their strong effects on software development effort estimates. J. Syst. Software (in press).

Maglio, S.J., Trope, Y., 2011. Scale and construal: how larger measurement units shrink length estimates and expand mental horizons. Psychon. Bull. Rev. 18 (1), 165–170.

Moløkken, K., Jørgensen, M., 2003. A review of software surveys on software effort estimation. International Symposium on Empirical Software Engineering, Rome, Italy. Simula Res. Lab, Lysaker, Norway, pp. 223–230.

Monga, A., Bagchi, R., 2012. Years, months, and days versus 1, 12, and 365: the influence of units versus numbers. J. Consum. Res. 39 (1), 185–198.

Moses, J., Farrow, M., 2003. A procedure for assessing the influence of problem domain on effort estimation consistency. Software Qual. J. 11 (4), 283–300.

Mussweiler, T., Strack, F., 2001. The semantics of anchoring. Organ. Behav. Hum. Decis. Processes 86 (2), 234–255.

Oppenheimer, D.M., 2008. The secret life of fluency. Trends Cognit. Sci. 12 (6), 237–241.

Prechelt, L., Unger, B., 2000. An experiment measuring the effects of personal software process (PSP) training. IEEE Trans. Software Eng. 27 (5), 465–472.

Rachlinski, J.J., Wistrich, A.J., Guthrie, C., 2015. Can judges make reliable numeric judgments: distorted damages and skewed sentences. Ind. LJ 90, 695.

Sherif, M., Hovland, C.I., 1961. Social judgment: Assimilation and contrast effects in communication and attitude change. Yale University Press, Oxford, England xii 218 pp.

Siddiqui, R.A., May, F., Monga, A., 2014. Reversals of task duration estimates: thinking how rather than why shrinks duration estimates for simple tasks, but elongates estimates for complex tasks. J. Exp. Soc. Psychol. 50, 184–189.

Song, H., Schwarz, N., 2008. If it's hard to read, it's hard to do processing fluency affects effort prediction and motivation. Psychol. Sci. 19 (10), 986–988.

Strack, F., Mussweiler, T., 1997. Explaining the enigmatic anchoring effect: mechanisms of selective accessibility. J. Pers. Soc. Psychol. 73 (3), 437.

Tamrakar, R., Jørgensen, M., 2012. Does the use of Fibonacci numbers in planning poker affect effort estimates?. In: 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012). IET, pp. 228–232.

Tausworthe, R.C., 1980. The work breakdown structure in software project management. J. Syst. Software 1 (3), 181–186.

Trope, Y., Liberman, N., 2010. Construal-level theory of psychological distance. Psychol. Rev. 117 (2), 440.

Ülkümen, G., Thomas, M., 2013. Personal relevance and mental simulation amplify the duration framing effect. J. Mark. Res. 50 (2), 194–206.

Ülkümen, G., Thomas, M., Morwitz, V.G., 2008. Will I spend more in 12 months or a year? The effect of ease of estimation and confidence on budget estimates. J. Consum. Res. 35 (2), 245–256.

van Genuchten, M., 1991. Why is software late? An empirical study of reasons for delay in software development. IEEE Trans. Software Eng. 17 (6), 582–590.

Yan, J., Hou, S., Unger, A., 2014. High construal level reduces overoptimistic performance prediction. Soc. Behav. Pers.: Int. J. 42 (8), 1303–1313.

Magne Jørgensen is a chief research scientist at Simula Research Laboratory, a professor at University of Oslo, an advisor at Scienta and a guest professor at Kathmandu University. His research includes work on management of software projects, evidence-based software engineering and human judgment. He has published more than 70 papers on these and other topics in software engineering, forecasting, project management and psychology journals. He has been ranked the top scholar in systems and software engineering four times and was in 2014 given the ACM Sigsoft award for most influential paper the last ten years for his work on evidence-based software engineering.