Trust yourself! Or maybe not: factors related to overconfidence and uncertainty assessments of software effort estimates

Patricia G. F. Matsubara
patriciagfm@icomp.ufam.edu.br
Institute of Computing (UFAM)
Manaus - AM, Brazil

Igor Steinmacher
Universidade Tecnológica Federal do Paraná – Campus Campo Mourão
Campo Mourão - PR, Brazil
igorfs@utfpr.edu.br

José Carlos Maldonado
University of São Paulo (ICMC-USP)
São Carlos - SP, Brazil
jcmaldon@icmc.usp.br

Bruno Gadelha
Institute of Computing (UFAM)
Manaus - AM, Brazil
bruno@icomp.ufam.edu.br

Tayana Conte
Institute of Computing (UFAM)
Manaus - AM, Brazil
tayana@icomp.ufam.edu.br

ABSTRACT

Software effort estimates are uncertain, given that they are probabilistic assessments of the future. Evaluating their uncertainty involves assigning them an appropriate confidence level and is paramount for satisfying commitments in software projects. However, estimators tend to be overconfident about their estimates, hampering the accuracy of their uncertainty assessments. Our research goal is to identify the factors related to overconfidence and uncertainty assessments in software estimation. To do so, we carried out a Systematic Literature Mapping (SLM), based on automated and snowballing searches. Our findings include eight factors related to overconfidence and uncertainty assessment. Some of them resulted in unexpected implications for practice. We also identified valuable and easy-to-use metrics that software practitioners can apply smoothly in their daily practice. Additionally, very few field and respondent studies exist about the topic. The software engineering area can significantly benefit from investigating how much practitioners know about the overconfidence effect, as well as of a better comprehension of the perceived importance, practices, and accuracy of uncertainty assessments in the software industry.

CCS CONCEPTS

• **Software and its engineering → Software development process management**.

KEYWORDS

Software effort estimation, Uncertainty assessments, Overconfidence

ACM Reference Format:
Patricia G. F. Matsubara, Igor Steinmacher, José Carlos Maldonado, Bruno Gadelha, and Tayana Conte. 2021. Trust yourself! Or maybe not: factors related to overconfidence and uncertainty assessments of software effort estimates. In *Brazilian Symposium on Software Engineering (SBES '21), September 27-October 1, 2021, Joinville, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474624.3474643

SBES '21, September 27-October 1, 2021, Joinville, Brazil

ACM ISBN 978-1-4503-9061-3/21/09…$15.00
https://doi.org/10.1145/3474624.3474643

1 INTRODUCTION

Software estimates are predictions about costs or durations [33]. Estimates are uncertain [9] because they are a probabilistic assessment of the future: one value in a range of possibilities with a particular probability of coming true [26]. Learning to assess the uncertainty of estimates is fundamental for the success of judgment-based estimation [13], the preferred estimation method in the software industry [3, 43, 45].

Software engineering practitioners are aware of the impact of uncertainty over their estimates, as reported by surveys [5, 30] and case studies [29, 31]. For instance, participants in one interview study reported that people disregard uncertainty by using single-point estimates instead of intervals and by treating estimates as promises [29]. A follow-up survey found out that expecting first estimates to correspond to actual values is a very common behavior regarding software estimates [30]. This suggests that the software industry would benefit from a more explicit uncertainty assessment process, aimed at accurately evaluating and informing uncertainty levels. Hence, it is relevant to understand the factors that impact uncertainty assessments.

One of such factors is the overconfidence effect: our tendency to believe our judgments, abilities and skills are superior than they objectively are [39, 44]. A review about judgment-based uncertainty assessments in software estimation revealed a systematic overconfidence, i.e., people are biased to underestimating uncertainty [13]. Such tendency is problematic, considering that we use our estimates as the basis for our plans [33]. Thus, it is also relevant to identify the factors contributing to the overconfidence phenomenon, because it is also related to people feeling excessively certain they know the truth [36].

Considering this, a useful piece of information for researchers and practitioners is the set of technical, social, or human factors affecting overconfidence and our uncertainty assessments. In search for them, we conducted a Systematic Literature Mapping (SLM) using the guidelines of Kitchenham, Budgen, and Brereton [27] and Petersen et al. [38]. Our research question is: "What are the factors related to overconfidence and the uncertainty assessment of judgment-based estimates?"

We examined the factors that we found to identify good practices to adopt and bad practices to avoid. We contribute to the existing literature through the identification and of eight factors, which resulted in a set of surprising implications for practice. For instance, the identification of more risks immediately prior to uncertainty assessments raises overconfidence and, therefore, should be avoided. We were also interested in understanding more about how researchers conducted their studies about the factors, both to summarize the current research status and to identify improvements for future investigations. Therefore, we contribute with the identification of the measurement strategies that researchers applied to understand the factors. Software practitioners can apply some of them smoothly in their daily practices due to their usefulness and easiness of use. Additionally, we mapped the research strategies the researchers employed the most, revealing new directions for future studies to increase the existing evidence's strength and guide the search for new factors.

2 BACKGROUND

Even though estimates are uncertain, we use them as the basis for activities that require a high degree of certainty, such as project planning, budgeting, or bidding. This is one of the features that differentiate software engineering from programming alone: software engineers need to make tough decisions with harder-stakes outcomes, frequently based on imprecise estimates [47]. In such contexts, knowing how uncertain estimates are is paramount to avoid missing relevant deadlines, overspending, or profit losses: we need to engage in uncertainty assessment.

To communicate the uncertainty of single-point estimates — an estimate presented as a single value — we should assign them a confidence level: a percentage that expresses the probability that the estimate equals the actual value [33]. A probabilistic view of estimates contrasts with a deterministic one, in which an estimate is purely a single value [7]. Figure 1.a illustrates single-point estimates, represented by the most likely effort (ML) and their confidence levels (CL). When estimates are not assigned a confidence level, people' interpretation about the meaning of the estimate vary largely, no matter whether they are communicating or receiving the information [16]. Software practitioners and project stakeholders can treat the estimate as representing the ideal effort, the most likely usage of effort, the median effort, or even a risk-averse effort – and all these interpretations have entirely different meanings.

An alternative is the adoption of prediction intervals [33], because hitting the target on single-point estimates is not realistic [10]. In fact, the use of single-point estimates instead of prediction intervals is a way to disregard the uncertainty of estimates [29]. We also illustrate a prediction interval in Figure 1.a, composed of a minimum effort (Min), the most likely usage of effort (ML), and the maximum effort (Max). Prediction intervals are a range we believe includes the actual value (AC) [9]. We also should assign them a confidence level, and wider intervals are typically associated with higher confidence levels.

Nevertheless, using prediction intervals can bring challenges. First, measuring their accuracy level is less evident compared with measuring it for single-point estimates [25]. The level of accuracy of single-point estimates is given by the average error no matter whether the estimate is too high or too low [9] and can be measured using metrics like the Magnitude of Relative Error (MRE) [21]. Nevertheless, these metrics require a single estimated value in their formulae, and prediction intervals have at least two values: a minimum and a maximum.

Second, there is the issue of the precision of a prediction interval, which is communicated through its width [48]: a narrower interval is more precise than a wider one. Figure 1.b illustrates the comparison between accuracy and precision in the case of prediction intervals. It shows two prediction intervals with the same width, that is, the same precision level: they have the same minimum (Min), most likely usage of effort (ML), and maximum effort (Max) values. Nevertheless, they have different accuracy levels. One of them is accurate, because the actual value (AC) is between the Min and the Max values. The other one is inaccurate because the AC is lower than the Min value (and outside the Min-Max range).

The precision versus accuracy issue in prediction intervals has been described as a trade-off between accuracy and informativeness: wider prediction intervals are considered less informative compared with narrower ones [48], even though they are more likely to include the actual values. On the one hand, humans tend to provide too narrow intervals for high confidence levels [12]. Such a situation is termed overprecision and is a variety of overconfidence [35]: our tendency to believe our judgments, abilities and skills are superior compared with what they objectively are [39, 44]. Overprecision is also the most pervasive and least understood form of overconfidence [36]. On the other hand, prediction intervals derived by formal models tend to be too wide to be informative, lowering their usefulness [12]. Therefore, existing guidelines on uncertainty assessments of software estimates suggest that the most promising strategies involve supporting judgment-based assessments instead of replacing them with formal models [12].

Moreover, the complete set of these guidelines were proposed by Jørgensen [12], and dates back to 2005. They are based on empirical studies from software cost estimation and other domains, such as psychology, forecasting, and decision making. Recently, Halkjelsvik and Jørgensen [9] made additional recommendations, reinforcing the need for appropriate framing (Guideline 7 from [12]). They added one recommendation regarding learning from accuracy feedback, which the original guidelines did not address.

Reviewing the most recent existing literature through an SLM, in this paper we contribute with eight factors related to overconfidence and uncertainty assessments of software estimates. In contrast, the latest previous literature reviews, like those by Basten and Sunyaev [4], Sehra et al. [40], and Carbonera, Farians, and Bischoff [6], focused on software effort estimation in general. We also discuss how researchers measured precision and accuracy in their studies and the research strategies they employed. We detail the method we used in the next section.

3 RESEARCH METHOD

Given that experts tend to be overconfident in their uncertainty assessments of software estimates [12], we aimed at investigating the factors that impact their overconfidence and their process of uncertainty assessment. Therefore, we defined the following research questions to guide our research:

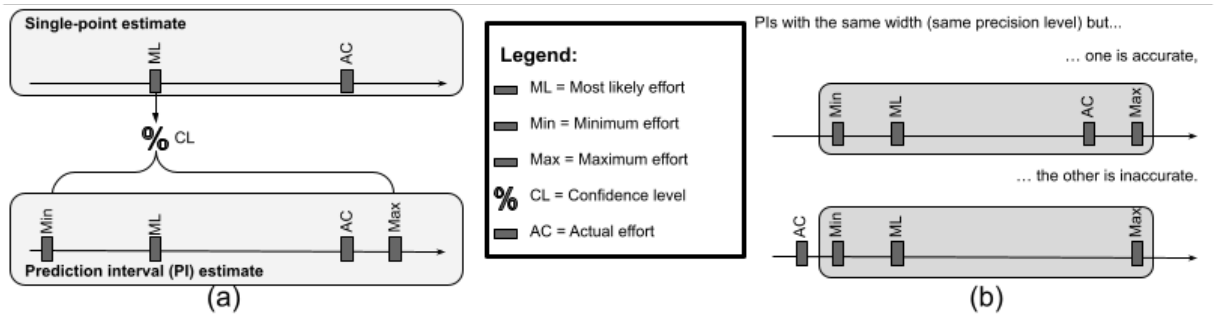

Figure 1: Single-point estimates and prediction intervals.

- RQ 1: What are the factors related to overconfidence and the uncertainty assessment of software estimates?
- RQ 2: How did researchers measure the relation of the factors to overconfidence or uncertainty assessments?
- RQ 3: What are the research strategies applied in the studies regarding the factors?

We focused on uncertainty assessments derived from expert judgment because it is the preferred estimation method in the software industry [34, 43, 45]. To answer the questions that we proposed, we executed an SLM based on the guidelines of Kitchenham, Budgen, and Brereton [27] and Petersen et al. [38]. We provide the details from the search, selection, extraction, and analysis in the next subsections.

3.1 Search and selection

We defined an oracle of papers we knew were relevant to our SLM by inspecting the references used in a set of guidelines [12]. We selected five papers that were empirical studies related to expert-judgment uncertainty assessments in the software engineering domain. Many other papers were not from the software engineering domain (such as from psychology), were reviews themselves, were related to uncertainty assessment derived from models, or were not specific about uncertainty assessments or overconfidence (but about estimation accuracy and errors). We present the papers in the oracle in Table 1.

We used the search string presented in Figure 2.a, deriving them from keywords from the papers in our oracle, an approach that many systematic mapping studies use [38]. We followed the steps recommended by Kitchenham, Budgen, and Brereton [27]: we reviewed our research questions and the title, abstract, and keywords from our oracle to identify relevant concepts and frequently used terms. We also relied on our domain knowledge and previous experiences [49]. Later, we evaluated our search string against our oracle [49].

We ran the automated search on Engineering Village, IEEE, and Scopus, as we illustrate in Figure 2.b. We found a total of 858 papers (Figure 2.b, Step 1). After removing duplicates, we had 538 papers (Figure 2.b, Step 2). Following, we read the title and abstracts of papers (Figure 2.b, Step 3), applying the inclusion (IC) and exclusion criteria (EC). We provide our IC and EC in Table 2. This procedure left us with 17 papers.

Paper title	Reference
Better sure than safe? Overconfidence in judgement based software development effort prediction intervals	[25]
Uncertainty intervals versus interval uncertainty: an alternative method for eliciting effort prediction intervals in software development projects	[24]
Realism in assessment of effort estimation uncertainty: it matters how you ask	[11]
Eliminating overconfidence in software development effort estimates	[23]
Combination of software development effort prediction intervals: why, when and how?	[22]

Table 1: Oracle.

Most papers from the automated search were removed because of EC01(46%) or because of EC02 (43%). That means they were either not about software estimation, they did not focus on overconfidence or uncertainty assessment, or they were about estimation methods other than judgment-based.

Next, we read the full text of the papers, resulting in 12 papers selected (Figure 2.b, Step 4). We also applied the same criteria from Table I. We evaluated our search string based on a goal to find at least 70% of the oracle papers, as Zhang et al. [49] recommends. We found four of them, which represents 80%, above our goal. Still, we considered the 12 papers resulting from the automated search as the seed for snowballing, getting additional 307 papers (Figure 2.b, Step 5), from which we selected two (Figure 2.b, Step 6). These two papers included the paper from the oracle that we missed during the automated search. Therefore, we had a final set of 14 papers selected (Figure 2.b, Step 7), including all the ones in the oracle.

3.2 Data extraction and analysis

We extracted the information presented in 3 from the papers. The template for the form, as well as the complete data extraction are provided as part of our supplementary material [32].

(a) Search string

(("uncertainty" OR "*confidence") AND ("effort estimation" OR "effort estimate" OR "cost estimation" OR "cost estimate" OR "duration estimation" OR "duration estimate" OR "schedule estimation" OR "schedule estimate" OR "size estimation" OR "size estimate")) AND ("software" OR "system"))

(b) Search and selection overview

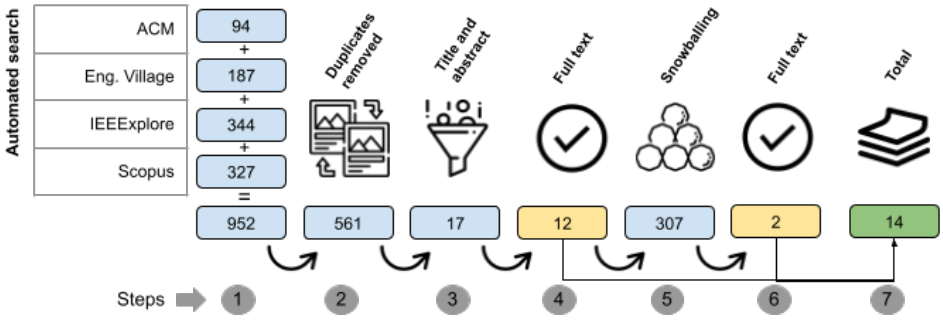

Figure 2: Search and selection overview.

ID	Criteria
IC 01	The paper reports one or more factors related to the overconfidence of estimators in the context of expert judgment estimation.
IC02	The paper reports one or more factors related to the uncertainty assessment of estimates in the context of expert judgment estimation.
EC01	The paper is not about software estimation, or it is about software estimation but does not focus on overconfidence or uncertainty assessment.
EC02	The paper is about software estimation other than judgment-based.
EC03	The paper is a literature review (systematic or not), lessons learned, or opinion paper and does not report empirical results regarding factors related to the overconfidence effect or the uncertainty assessment of estimates.
EC04	The paper is about expert judgment estimation and even cites the overconfidence effect or the uncertainty assessment of estimates, but does not report any related factor.
EC05	The paper presents non-peer-reviewed results.
EC06	The paper is not written in English.
EC07	The paper is not accessible in full-text online.
EC08	The study is published as a book or grey literature.
EC09	The paper is a duplicate or a previous version of another already selected paper.
EC010	The paper does not describe the factors to allow for categorization.

Table 2: Inclusion and exclusion criteria sample.

After the data extraction, we produced one or more codes for all the text fragments we extracted as part of the "factor and discussion"

Information	RQ
Paper title, authors, venue, venue type (conference or journal), year of publication	-
Context: a brief explanation of the study and participants. Factor descriptions and definitions.	-
Factors and discussion: including text about the research results and the researchers' discussion about them.	1
Measurement strategy used to evaluate the relation of the factor	2
The research strategy that the researchers adopted, according to the classification of Storey et al. [42]	3

Table 3: Data extraction.

section of the extraction form. We generated the codes obeying to the following structure: <candidate factor> + <a brief description of effects>. The candidate factor was the label that the original study' authors provided. The description of effects highlighted whether the factor led to overconfidence, accuracy improvement, etc. We reread all the codes after this step to identify similarities in concepts. We aggregated similar candidate factors under a final factor label chosen to reflect their core. Three authors held regular meetings to review the factors and codes. The complete set of codes is also available in our supplementary material [32].

4 RESULTS AND DISCUSSION

In this section, we answer and discuss the results to each research question, linking the factors to overconfidence and/or to uncertainty assessment, as we illustrate in Figure 3, which summarizes all our results together. When applicable, we also provide recommendations for practitioners and researchers. We start with the factors (RQ 1) in Section 4.1. Next, we discuss the measurement strategies (RQ 2) in Section 4.2. Then, we discuss the research strategies in Section 4.3.

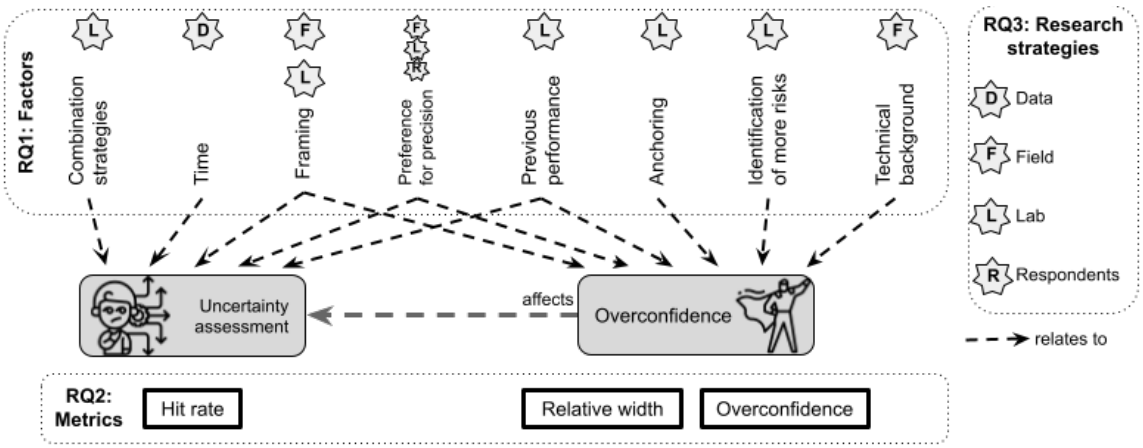

Figure 3: Factors, metrics, and research strategies.

4.1 RQ 1: Factors

In 4, we present all the factors and their respective papers. In our context, a factor is any element that contributes to overconfidence or is related to the uncertainty assessment of software estimates. We found a total of eight factors supported by the 14 papers that we included in our SLM: two more closely related to uncertainty assessments, three to overconfidence, and three to both.

Factor	Papers
Anchoring effect	[2]
Combination strategies	[22]
Framing	[11, 17, 19, 24]
Identification of more risks	[15]
Preference for precision over accuracy	[18, 25]
Previous performance	[8, 20, 21, 23]
Technical background	[25]
Time	[28]

Table 4: List of factors.

Regarding uncertainty assessments, the combination strategies based on group discussions are superior to other strategies. They are used when combining individual prediction intervals to define a final one based on the individual minimum and maximum values. Jørgensen and Moløkken [22] investigated three combination strategies for deriving the final prediction interval: (i) widest interval, using the lowest of the individual minimum values and the highest of the individual maximum values; (ii) team (or group discussion), using group discussions to define the final prediction interval; and (iii) average, using the average of the individual minimal values and the average of the individual maximum. The group discussion and widest interval strategies led to more accurate prediction intervals. Additionally, the team strategy achieved comparable accuracy to the widest interval strategy but with narrower prediction intervals, which shows it is more efficient [22].

Surprisingly, we found no additional studies about the combination strategies of prediction intervals since the publication of the existing guidelines ([12]). Therefore, the evidence we have about it comes from one paper only, using a laboratory research strategy. Given the relevance and efficacy of combining individual estimates to improve accuracy instead of relying upon single sources [14], it seems prudent to look for more evidence about the different combination strategies, preferably using other research strategies, such as field studies. In the lack of more and recent evidence, the combination of prediction intervals through group discussions is still the best recommendation for improving uncertainty assessments.

The other factor related to uncertainty assessments is time passage. The evidence, coming from one data study, shows that time alone—which is supposed to involve more knowledge gains—was unrelated to improvements regarding the uncertainty of estimates, as we might expect from the idea of the cone of uncertainty [28]. Given this contraction with the suggestions from past literature, this factor could benefit from further research studies. In any case, we cannot expect time alone to reduce the uncertainty of our estimates, given the evidence we found.

Aranda and Easterbrook [2] showed that the anchoring effect affects prediction intervals. This effect is a product of our tendency to base our estimates on an initial value (the anchor value) and then adjust, although we usually do not adjust enough [39]. It is one of the three factors related exclusively to overconfidence, as we show in 3. Although only one paper in our set investigates it through a laboratory study, this effect is also reported as affecting software estimates in general [4], not only overconfidence, reinforcing the strength of the evidence about it. So far, the best strategy to eliminate the anchoring effect is precaution: avoid exposing estimators to anchor values [41].

Another factor related to higher overconfidence is the identification of more risks prior to estimation, , according to the results of four experiments [15] — a result that we found startling. Indeed, the estimation literature even recommends using risk quantification, which involves risk identification, too, as part of the process of communicating the uncertainty of estimates [33]. However, the researchers reported that they did not investigate a complete risk analysis and management process. Also, a higher illusion of control might explain the lack of reduction in overconfidence in this case

[15]. In any case, more research about this can clarify the issue, specially to increase the generalizability and realism of the results, since the current evidence comes from a laboratory study alone. In any case, it is wise to avoid risk identification immediately prior to uncertainty assessments, to avoid higher overconfidence.

Likewise, the technical background of a role—which supposedly indicates people are more knowledgeable about software estimation—did not lead to better effort prediction intervals, only to higher confidence in estimates [25]. This result comes from a field study, indicating the realism of the findings. So, we cannot expect that people with a technical background will improve uncertainty assessments due to reduced overconfidence — perhaps it is the other way around: we also need people with non-technical background during estimation activities.

Three factors were related both to overconfidence and uncertainty assessments. One of them is framing, investigated through laboratory and field studies, related to different variations of a traditional and an alternative format to ask for uncertainty assessments [11, 17, 19, 24]. Using the traditional format, estimators are asked to provide a prediction interval that they believe is likely to include the actual estimate with a given confidence level. When using an alternative format, estimators have to assess the probability of including the actual value in a fixed prediction interval calculated from the most likely estimate (like the interval where the minimum value is defined as 50% of the most likely estimate and the maximum value is defined as 200% of the most likely estimate) [11, 19, 24]. Also, in the traditional format, researchers found that asking for prediction intervals with significantly different confidence levels does not lead to significantly different prediction intervals, as expected [17]. All existing evidence point to superiority of framing requests by using an alternative format compared to the traditional one: it provides more accurate uncertainty assessment, as the original guidelines preached [12].

A preference for precision over accuracy also affect uncertainty assessments and is related to overconfidence. Precision, in the form of narrower prediction intervals, makes estimators look like more skilled [25] and competent [19], and estimates look more trustworthy [18]. Even though we should expect people to assess wider intervals as more accurate because they are more likely to include the actual values, they see narrower prediction intervals as more accurate even when the interval does not include the actual effort, compared to wider effort intervals, even when the wider interval includes the actual effort [18]. Additionally, wider prediction intervals are considered less acceptable than lower confidence levels [25]. The preference for precision was the factor investigated using the most varied set of research strategies: field, laboratory, and respondent studies. This indicates the realism, researchers' control, and generalizability of this finding.

The preference for precision over accuracy can also help us to understand more our results regarding framing: wider prediction intervals are less acceptable than lower confidence levels [25]. The alternative format asks the estimator to give a confidence level, fixating the prediction interval – while the traditional format does precisely the opposite.

Making estimators recall or giving them feedback about the previous performance of other tasks and projects also reduces overconfidence [20] and improves accuracy [8, 23]. Estimators who were asked to recall their past effort usage were less overconfident than others who were not asked to do it [20]. In another study, those asked to remember the historical estimation error distribution of previous projects were more accurate when estimating the minimum values of prediction intervals. However, they were not more accurate regarding maximum values [23]. The use of more explicit assessment strategies of previous performance also led to more realistic uncertainty assessments [8]. One of these strategies included the adjustment of confidence levels based on the hit rate of previous reasonably similar tasks. Another explicit strategy is the recall of the most similar tasks and the use of the estimation error of those tasks to determine the confidence of the current task. Nevertheless, in one study, the researchers found that structured lessons learned sessions about previous software estimation performance led to no decrease in overconfidence neither improvements in uncertainty assessments [21]. However, such sessions involved more than recall and feedback on estimation error and realism of uncertainty assessment of previous tasks. They also involved a learning moment, when the estimators had to reflect on perceived reasons for their estimation performance, including the realism of their confidence levels and suggesting lessons learned. This last result is surprising because we expected that more information and more reflection led to improved decision-making and could be investigated deeper to assess its realism.

Implications for practice:

- Frame your requests for uncertainty assessments by fixating the prediction interval and asking for confidence levels.
- Avoid exposing estimators to anchor values.
- Avoid asking for uncertainty assessments immediately after risk identification.
- Add one step for recalling previous estimation performance during uncertainty assessments. Provide feedback on uncertainty assessment accuracy.
- Combine prediction intervals using group discussions.
- Educate software teams and other stakeholders on the relationship between prediction interval width, accuracy, and precision.
- Do not expect people with more technical backgrounds will provide better uncertainty assessment or be less overconfident.
- Do not expect time passage alone to reduce the uncertainty of estimates.

Implications for research:

- There is a need to gather more evidence for the different combination strategies of individual prediction intervals.
- There is a need to clarify the contexts and situations when more information and reflection are beneficial, when they are not, and why.

4.2 RQ 2: Measurement strategies

In Table 5 we show the distribution of studies according to the different strategies that researchers used to measure the relationship between the factors and overconfidence or uncertainty assessments.

Measurement strategy	Papers
Hit rate	[22, 24, 25]
Overconfidence (or a variation)	[8, 11, 20, 21]
Relative width	[17, 19, 22, 25]
Participants' perceptions	[18, 25]
Difference of prediction intervals	[2, 23]
Width-accuracy correlations	[11, 25]
Other	[11, 15, 19, 25, 28]

Table 5: List of measurement strategies.

Researchers have employed a wide variety of measurement strategies to understand factors related to overconfidence and uncertainty assessments. We found little agreement, with overconfidence and relative width measures being the most adopted ones. Also, there are different measures providing researchers with diverse perspectives about the factors. For instance, while hit rate is a measure of uncertainty assessment accuracy, the relative width can provide information about the precision, which is related to overconfidence.

Hit rate is the proportion of prediction intervals that include the actual usage of effort [22]. We illustrate it in Figure 4, which has four prediction intervals: A, B, C, and D. Estimators provided them with a CL of 90%. Nevertheless, only the intervals B and D included the actual values (AC). Therefore, the hit rate is 50%, much lower than the CL. The hit rate provides us a simple measure of accuracy for prediction intervals, aiding us in evaluating our uncertainty assessments [25], as we showed in 3. We can expect the hit rate to correspond to the confidence level of prediction intervals in the long run [25]. It is a simple to use and understand measurement strategy.

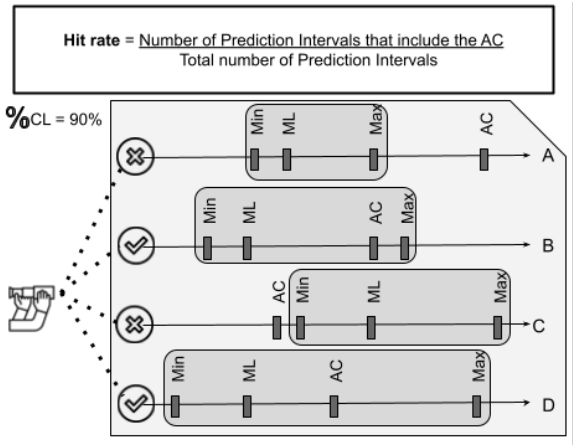

Figure 4: Hit rate metric.

The level of correspondence between the confidence level associated with prediction intervals and the hit rate gives us a measure of overconfidence, as we show in Figure 5. It is easily calculated by subtracting the hit rate from the confidence level. A positive level represents overconfidence. Nevertheless, the researchers use slightly different metrics and names for the construct. One of them is the overconfidence metric, calculated as: OVERCONFIDENCE = AVERAGE CONFIDENCE LEVEL - HIT RATE [8, 20]. In one study, the researchers changed the formula to: OVERCONFIDENCE = (AVERAGE CONFIDENCE LEVEL - HIT RATE)/HIT RATE [21]. In another study, the researchers inverted the overconfidence formula terms and called the new measure of bias: BIAS = HIT RATE - CONFIDENCE LEVEL [11].

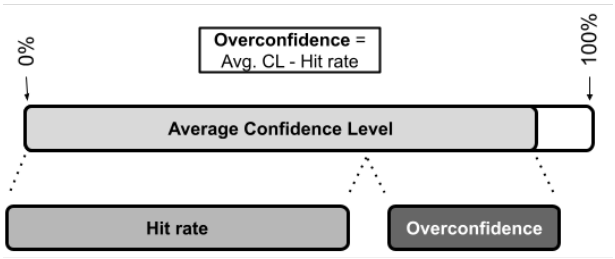

Figure 5: Overconfidence metric.

The relative width is a measure of precision illustrated in Figure 6. It is calculated as: RELATIVE WIDTH = (MAXIMUM – MINIMUM)/MOST LIKELY ESTIMATE [17, 19, 22, 25]. It may be harder to interpret in practice because, at first sight, it might not be clear that a higher value means less precision than a lower one. In other words, a too low relative width indicates overprecision, a variety of overconfidence [36]. Also, achieving a high hit rate (high accuracy) maintaining a low relative width (high precision) is desirable, and the combination of the two strategies has been used to evaluate the efficiency of prediction intervals [22].

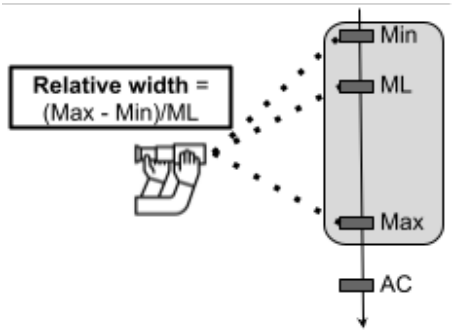

Figure 6: Relative width metric.

Researchers also resorted to the participants' perceptions as receivers of estimates, to understand more about their interpretation of prediction intervals. The participants had to assess the competence of estimators and the trustworthiness of their estimates given different levels of precision of prediction intervals that these

estimators provided [18, 25]. This revealed that overconfidence is not universally condemned [25]. On the contrary, previous studies about the use of checklists in software estimation, for instance, praised such artifacts for increasing the confidence of estimators [37, 46]. At the same time, the papers included in our SLM looked for strategies to reduce it because they consider we are naturally overconfident.

Researchers also analyzed the difference in prediction intervals. For instance, to understand the impact of anchors, they compared the mean of the maximum values in a group exposed to a low-value anchor to the mean of the minimum values in a group exposed to a high-value anchor [2]. Also, to understand the impact of recalling historical data errors, the researchers compared the averages of minimum and maximum values of the prediction intervals of the two participating groups [23].

An additional measurement strategy is the use of width-accuracy correlations, which involves calculating the correlation between a prediction interval relative width with an accuracy measure, such as MRE (Magnitude of Relative Error) and BRE (Balanced Relative Error). The correlation is positive if the estimators base their prediction interval on extensive knowledge about the uncertainty of the estimated tasks and if different tasks have different levels of uncertainty [25].

Finally, we found a few measurement strategies used in one study each, classifying them under the label of "other", such as measuring the ARPI (Actual effort Relative to Prediction Interval), that defines the distance between the actual effort and the midpoint of the prediction interval, normalized by the prediction interval width [25]. The complete set of metrics we classified under the label of "other" is presented in our supplementary material [32].

We also generated a template for the hit rate, overconfidence, and relative width, that researchers and practitioners can use to calculate them, and is part of our supplementary material [32]. Additionally, the plethora of measurement strategies is a sign of the complexity of the problem of uncertainty assessments and overconfidence.

Implications for practice and research:

- There are simple to understand measurement strategies we can use to evaluate uncertainty assessments and overconfidence, like hit rate and the overconfidence metric.
- Whenever possible, multiple measurement strategies are preferable because each metric provides a different angle of the complex problem of uncertainty assessments and overconfidence.

4.3 RQ 3: Research strategies

Table 6 presents the distribution of papers for each research strategy defined by Storey et al [42]. Lab studies are conducted in contrived settings, involving hypotheses testing in controlled situations. Field studies are executed in the context of work, with researchers entering a natural setting to study a phenomenon or a system in action. Respondent studies gather insights from practitioners and other stakeholders, in settings of convenience. Data strategies rely on archival, generated, and simulated data.

Research strategy	# of papers	Papers
Lab	11	[2, 8, 15, 17, 19–25]
Field	2	[11, 25]
Respondents	2	[18, 25]
Data	1	[28]

Table 6: Distribution of papers per research strategy.

The distribution of our primary studies is very unbalanced, and most studies adopted a lab strategy (11 occurrences) , evaluating one factor only in a controlled setting through hypothesis testing [42]. Few studies adopted field (two occurrences) or respondents (two occurrences) strategy. Only one study adopted the data strategy. Additionally, only one paper reported using multiple strategies [25] to understand the investigated factors from different perspectives.

Therefore, an implication for research is to diversify the research strategies employed in future studies investigating the factors related to overconfidence and uncertainty assessments. It would be particularly interesting to investigate how much practitioners know about the overconfidence effect, given that previous research indicates they are not clear about what they mean with their estimates: whether they are ideal or risk-averse estimates, for instance [16]. Additionally, we found no surveys — a type of respondent study — about the uncertainty assessment perceived importance and practices in the software industry. More data strategies also enrich the knowledge of the accuracy of uncertainty assessments in the industry.

Implications for research:

- There is a need to investigate the factors in the field and through respondent studies to increase the realism and to understand how generalizable the results are.
- It is interesting to investigate the knowledge of software practitioners about the overconfidence effect, its impact on software estimation, and software projects' success.
- There is a need to investigate more about the perceived importance, practices, and accuracy of uncertainty assessments in software projects in the industry.

5 THREATS TO VALIDITY

One of the threats for study selection validity is the adequacy of initial relevant publications identification, addressed with an automatic search in known digital libraries, guided by a known set of papers [1]. We used the known set of papers to evaluate the search strategy [49]. We also used snowballing procedures to identify additional relevant papers. Another threat to study selection validity for this SLM is the study inclusion/exclusion bias, addressed through the definition of study inclusion and exclusion criteria in the research protocol.

Another threat is the bias of classification schema [1]. To avoid it, we relied on previous existing classifications when possible, such as the research strategies framework of Storey et al. [42]. We aggregated similar findings under labels, and the authors held

meetings to review the factors and codes. We also compared our factors to the guidelines of a previous published related work [12]. As for research validity, there is the threat of lack of repeatability [1]. We involved more than one researcher during the process to mitigate it and made all the SLM data publicly available.

6 CONCLUSION

In this paper, we presented eight factors related to the overconfidence of estimators and uncertainty assessment of estimates. The factors that researchers explored the most were the framing of requests for uncertainty assessments, previous performance in software estimation, and the preference for precision over accuracy. Some of the factors represent unexpected results, such as the fact that the identification of more risks immediately before software estimation can increase the overconfidence of estimators. Also, the technical background of a role does not lead to better effort prediction intervals, only to higher confidence in estimates, something that is surprising. Additionally, the uncertainty of estimates is not reduced by itself as the project progresses.

We also identified the measurement strategies that researchers used to understand more about such factors. Some of them are easy to understand and can be used in industrial practice instead of remaining in the academic context. Additionally, we discovered that most of the research papers we included used a lab research strategy to investigate the factors, revealing the need to understand more about the realism and generalizability of the results.

We contribute with the existing literature on overconfidence and uncertainty assessments providing implications for practice based on the factors we found on the empirical primary studies that we included in our SLM. Additionally, we contribute with the description of the measurement strategies. We also provide implications for research, regarding underexplored factors and research strategies.

Future research can include gathering more evidence regarding underexplored factors, such as the combination strategies of individual prediction intervals, or factors investigated using one kind of research strategy only, such as the use of previous performance. Additionally, the software engineering community benefits from knowledge about practitioners' awareness of the overconfidence effect and its impact on software estimation and the success of software projects. It would also be interesting to investigate the perceived importance, practices, and accuracy of uncertainty assessments in the software industry. Therefore, we plan a survey to understand more of the state-of-the-practice as future work. Another action is to introduce the metrics we found in software companies interested in improving their uncertainty assessments to evaluate their impact.

ACKNOWLEDGMENTS

This research, carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 48 of Decree no 6.008/2006(SUFRAMA), was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law no 8.387/1991, through agreement 001/2020, signed with Federal University of Amazonas and FAEPI, Brazil and through agreement no 003/2019 (PROPPGI), signed with ICOMP/UFAM. Also supported by CAPES - Financing Code 001, CNPq processes 314174/2020-6 and 313067/2020-1, and FAPEAM process 062.00150/2020.

REFERENCES

[1] Apostolos Ampatzoglou, Stamatia Bibi, Paris Avgeriou, Marijn Verbeek, and Alexander Chatzigeorgiou. 2019. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Information and Software Technology* 106 (Feb. 2019), 201–230. https://doi.org/10.1016/j.infsof.2018.10.006

[2] Jorge Aranda and Steve Easterbrook. 2005. Anchoring and Adjustment in Software Estimation. In *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE-13)*. ACM, New York, NY, USA, 346–355. https://doi.org/10.1145/1081706.1081761 event-place: Lisbon, Portugal.

[3] Tharwon Arnuphaptrairong. 2018. The State of Practice of Software Cost Estimation: Evidence From Thai Software Firms. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 2. Hong Kong, 6.

[4] Dirk Basten and Ali Sunyaev. 2014. A Systematic Mapping of Factors Affecting Accuracy of Software Development Effort Estimation. *Communications of the Association for Information Systems* 34 (2014). https://doi.org/10.17705/1CAIS.03404

[5] Ricardo Britto, Emilia Mendes, and Jürgen Börstler. 2015. An Empirical Investigation on Effort Estimation in Agile Global Software Development. In *2015 IEEE 10th International Conference on Global Software Engineering*. 38–45. https://doi.org/10.1109/ICGSE.2015.10 ISSN: 2329-6313.

[6] Carlos Eduardo Carbonera, Kleinner Farias, and Vinicius Bischoff. 2020. Software development effort estimation: a systematic mapping study. *IET Software* 14, 4 (2020), 328–344. https://doi.org/10.1049/iet-sen.2018.5334 _eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-sen.2018.5334.

[7] Stein Grimstad, Magne Jørgensen, and Kjetil Moløkken-Østvold. 2006. Software effort estimation terminology: The tower of Babel. *Information and Software Technology* 48, 4 (April 2006), 302–310. https://doi.org/10.1016/j.infsof.2005.04.004

[8] Tanja M. Gruschke and Magne Jørgensen. 2008. The role of outcome feedback in improving the uncertainty assessment of software development effort estimates. *ACM Transactions on Software Engineering and Methodology* 17, 4 (Aug. 2008), 20:1–20:35. https://doi.org/10.1145/13487689.13487693

[9] Torleif Halkjelsvik and Magne Jørgensen. 2018. *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*. Springer International Publishing. https://doi.org/10.1007/978-3-319-74953-2

[10] Jo Erskine Hannay, Hans Christian Benestad, and Kjetil Strand. 2019. Agile Uncertainty Assessment for Benefit Points and Story Points. *IEEE Software* 36, 4 (July 2019), 50–62. https://doi.org/10.1109/MS.2018.2875845 Conference Name: IEEE Software.

[11] M. Jorgensen. 2004. Realism in assessment of effort estimation uncertainty: it matters how you ask. *IEEE Transactions on Software Engineering* 30, 4 (April 2004), 209–217. https://doi.org/10.1109/TSE.2004.1274041 Conference Name: IEEE Transactions on Software Engineering.

[12] M. Jorgensen. 2005. Evidence-based guidelines for assessment of software development cost uncertainty. *IEEE Transactions on Software Engineering* 31, 11 (Nov. 2005), 942–954. https://doi.org/10.1109/TSE.2005.128 Conference Name: IEEE Transactions on Software Engineering.

[13] M. Jorgensen. 2005. Practical guidelines for expert-judgment-based software effort estimation. *IEEE Software* 22, 3 (May 2005), 57–63. https://doi.org/10.1109/MS.2005.73

[14] Magne Jorgensen. 2014. What We Do and Don't Know about Software Development Effort Estimation. *IEEE Software* 31, 2 (2014), 37–40. https://doi.org/10.1109/MS.2014.49

[15] Magne Jørgensen. 2010. Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates. *Information and Software Technology* 52, 5 (May 2010), 506–516. https://doi.org/10.1016/j.infsof.2009.12.002

[16] Magne Jørgensen. 2014. Communication of Software Cost Estimates. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14)*. ACM, New York, NY, USA, 28:1–28:5. https://doi.org/10.1145/2601248.2601262 event-place: London, England, United Kingdom.

[17] Magne Jørgensen. 2014. The Ignorance of Confidence Levels in Minimum-Maximum Software Development Effort Intervals. *Lecture Notes on Software Engineering* 2, 4 (2014), 327–330. https://doi.org/10.7763/LNSE.2014.V2.144

[18] Magne Jørgensen. 2016. The Use of Precision of Software Development Effort Estimates to Communicate Uncertainty. In *Software Quality. The Future of Systems and Software Development (Lecture Notes in Business Information Processing)*, Dietmar Winkler, Stefan Biffl, and Johannes Bergsmann (Eds.). Springer International Publishing, Cham, 156–168. https://doi.org/10.1007/978-3-319-27033-3_11

[19] M. Jørgensen. 2018. Looking Back on Previous Estimation Error as a Method to Improve the Uncertainty Assessment of Benefits and Costs of Software Development

Projects. In *2018 9th International Workshop on Empirical Software Engineering in Practice (IWESEP)*. 19–24. https://doi.org/10.1109/IWESEP.2018.00012 ISSN: 2573-2021.

[20] Magne Jørgensen, Bjørn Faugli, and Tanja Gruschke. 2007. Characteristics of software engineers with optimistic predictions. *Journal of Systems and Software* 80, 9 (Sept. 2007), 1472–1482. https://doi.org/10.1016/j.jss.2006.09.047

[21] Magne Jørgensen and Tanja M. Gruschke. 2009. The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments. *IEEE Transactions on Software Engineering* 35, 3 (May 2009), 368–383. https://doi.org/10.1109/TSE.2009.2

[22] Magne Jørgensen and Kjetil Moløkken. 2002. Combination of software development effort prediction intervals: why, when and how?. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE '02)*. ACM, Ischia, Italy, 425–428. https://doi.org/10.1145/568760.568833

[23] Magne Jørgensen and Kjetil Moløkken. 2004. Eliminating Over-Confidence in Software Development Effort Estimates. In *Product Focused Software Process Improvement (Lecture Notes in Computer Science)*, Frank Bomarius and Hajimu Iida (Eds.). Springer, Berlin, Heidelberg, 174–184. https://doi.org/10.1007/978-3-540-24659-6_13

[24] Magne Jørgensen and Karl-Halvor Teigen. 2002. Uncertainty Intervals Versus Interval Uncertainty: an Alternative Method for Eliciting Effort Prediction Intervals in Software Development Projects. Singapore, 343–352.

[25] Magne Jørgensen, Karl Halvor Teigen, and Kjetil Moløkken. 2004. Better sure than safe? Over-confidence in judgement based software development effort prediction intervals. *Journal of Systems and Software* 70, 1 (Feb. 2004), 79–93. https://doi.org/10.1016/S0164-1212(02)00160-7

[26] Barbara Kitchenham and Stephen Linkman. 1997. Estimates, Uncertainty, and Risk. *IEEE Software* 14, 3 (May 1997), 69–74. https://doi.org/10.1109/52.589239

[27] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. 2015. *Evidence-Based Software Engineering and Systematic Reviews* (1 ed.). CRC Press.

[28] T. Little. 2006. Schedule estimation and uncertainty surrounding the cone of uncertainty. *IEEE Software* 23, 3 (May 2006), 48–54. https://doi.org/10.1109/MS.2006.82 Conference Name: IEEE Software.

[29] Ana Magazinius, Sofia Börjesson, and Robert Feldt. 2012. Investigating intentional distortions in software cost estimation – An exploratory study. *Journal of Systems and Software* 85, 8 (Aug. 2012), 1770–1781. https://doi.org/10.1016/j.jss.2012.03.026

[30] A. Magazinius and R. Feldt. 2011. Confirming Distortional Behaviors in Software Cost Estimation Practice. In *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*. 411–418. https://doi.org/10.1109/SEAA.2011.61

[31] Ana Magazinovic and Joakim Pernstål. 2008. Any other cost estimation inhibitors?. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM '08*. ACM Press, Kaiserslautern, Germany, 233. https://doi.org/10.1145/1414004.1414042

[32] Patricia Matsubara, Igor Steinmacher, José Carlos Maldonado, Bruno Gadelha, and Tayana Conte. [n.d.]. Supplementary material for Trust Yourself! https://www.doi.org/10.6084/m9.figshare.14569230

[33] Steve McConnell. 2006. *Software Estimation: Demystifying the Black Art* (1 ed.). Microsoft Press, Redomnd, Washington, USA.

[34] K. Molokken and M. Jorgensen. 2003. A review of software surveys on software effort estimation. In *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.* 223–230. https://doi.org/10.1109/ISESE.2003.1237981

[35] Don A. Moore and Paul J. Healy. 2008. The trouble with overconfidence. *Psychological Review* 115, 2 (2008), 502–517. https://doi.org/10.1037/0033-295X.115.2.502 Place: US Publisher: American Psychological Association.

[36] Don A. Moore and Derek Schatz. 2017. The three faces of overconfidence. *Social and Personality Psychology Compass* 11, 8 (2017), 1–12. https://doi.org/10.1111/spc3.12331 Place: United Kingdom Publisher: Wiley-Blackwell Publishing Ltd.

[37] Ursula Passing and Martin Shepperd. 2003. An Experiment on Software Project Size and Effort Estimation. In *Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE '03)*. IEEE Computer Society, Washington, DC, USA, 120–. http://dl.acm.org/citation.cfm?id=942801.943632

[38] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (Aug. 2015), 1–18.

[39] Valerie F. Reyna, Priscila G. Brust-Renck, and Rebecca B. Weldon. 2021. Judgment and Decision Making. https://doi.org/10.1093/acrefore/9780190236557.013.536 ISBN: 9780190236557.

[40] Sumeet Kaur Sehra, Yadwinder Singh Brar, Navdeep Kaur, and Sukhjit Singh Sehra. 2017. Research patterns and trends in software effort estimation. *Information and Software Technology* 91 (Nov. 2017), 1–21. https://doi.org/10.1016/j.infsof.2017.06.002

[41] Martin Shepperd, Carolyn Mair, and Magne Jørgensen. 2018. An Experimental Evaluation of a De-biasing Intervention for Professional Software Developers. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC '18)*. ACM, New York, NY, USA, 1510–1517. https://doi.org/10.1145/3167132.3167293 event-place: Pau, France.

[42] Margaret-Anne Storey, Neil A. Ernst, Courtney Williams, and Eirini Kalliamvakou. 2020. The who, what, how of software engineering research: a socio-technical framework. *Empirical Software Engineering* 25, 5 (Sept. 2020), 4097–4129. https://doi.org/10.1007/s10664-020-09858-z

[43] Adam Trendowicz, Jürgen Münch, and Ross Jeffery. 2011. State of the Practice in Software Effort Estimation: A Survey and Literature Review. In *Software Engineering Techniques (Lecture Notes in Computer Science)*, Zbigniew Huzar, Radek Koci, Bertrand Meyer, Bartosz Walter, and Jaroslav Zendulka (Eds.). Springer Berlin Heidelberg, 232–245.

[44] Susan Tyler. 2020. Chapter 1: How We Use Our Expectations. In *Human Behavior and the Social Environment I*. University of Arkansas Libraries. https://uark.pressbooks.pub/hbse1/chapter/chapter-1/

[45] Muhammad Usman, Emilia Mendes, and Jürgen Börstler. 2015. Effort estimation in agile software development: a survey on the state of the practice. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE '15)*. Association for Computing Machinery, Nanjing, China, 1–10. https://doi.org/10.1145/2745802.2745813

[46] Muhammad Usman, Kai Petersen, Jürgen Börstler, and Pedro Santos Neto. 2018. Developing and using checklists to improve software effort estimation: A multi-case study. *Journal of Systems and Software* 146 (Dec. 2018), 286–309. https://doi.org/10.1016/j.jss.2018.09.054

[47] Titus Winters, Tom Manshreck, and Hyrum Wrigth. 2020. What is Software Engineering? In *Software Engineering at Google* (1st ed.). O'Reilly Media, Inc.

[48] Ilan Yaniv and Dean P. Foster. 1995. Graininess of judgment under uncertainty: An accuracy-informativeness trade-off. *Journal of Experimental Psychology: General* 124, 4 (1995), 424–432. https://doi.org/10.1037/0096-3445.124.4.424 Place: US Publisher: American Psychological Association.

[49] He Zhang, Muhammad Ali Babar, and Paolo Tell. 2011. Identifying relevant studies in software engineering. *Information and Software Technology* 53, 6 (June 2011), 625–637. https://doi.org/10.1016/j.infsof.2010.12.010