SEXTAMT: A systematic map to navigate the wide seas of factors affecting expert judgment software estimates

Patrícia Gomes Fernandes Matsubara[a, b, *, 1], Bruno Freitas Gadelha[a], Igor Steinmacher[c, d], Tayana Uchôa Conte[a]

[a] Institute of Computing of the Federal University of Amazonas (UFAM)
[b] Faculty of Computing of the Federal University of Mato Grosso do Sul (UFMS)
[c] Federal University of Technology – Paraná (UTFPR)
[d] Northern Arizona University (NAU)

ABSTRACT

Context: Software projects involve technical and managerial activities, including software estimation. Inaccurate estimates are harmful and improving estimation methods is not enough: we need to understand more of the factors that impact estimates. **Objective:** Our study aims to identify the existing evidence about the factors that affect estimates in software projects when using expert judgment. **Method:** We executed a Systematic Literature Mapping (SLM) based on database and snowballing searches, selecting papers by first reading their titles and abstracts and later reading the full text. **Results**: Researchers investigated a wide range of different factors employing mostly laboratory research strategies and relying primarily on differences of estimates and participants' perceptions to measure the factors' effects. Resulting from our analysis, we present the SEXTAMT (Software Estimates of eXperts: A Map of influencing facTors), a map of factors affecting estimates built on three dimensions: project/iteration phase, stakeholders, and type of effect. **Conclusion:** Over the years, researchers have investigated a varied set of factors. Many of them were explored in different studies, employing diverse research strategies. Such studies provide compelling evidence on the elements that influence expert judgment estimates, which can be used to assess and improve everyday estimation in the software industry.

Keywords: Expert judgment, Software effort estimation

[1] *Corresponding author
E-mail addresses: patriciagfm@icomp.ufam.edu.br, bruno@icomp.ufam.edu.br, igorfs@utfpr.edu.br, tayana@icomp.ufam.edu.br

1. Introduction

An estimate is a quantitative assessment of a variable's likely outcome, such as project costs, resources, effort, or duration [164]. Estimating tasks and projects is a critical part of developing and maintaining software, and researchers devoted a significant amount of effort to creating and assessing software estimation methods [79]. One such method is expert judgment: it is the preferred estimation method in the industry [121,149]. In agile software development, Planning Poker — based on expert judgment — is the most applied method [153]. Expert judgment is also on the rise as a research topic in software effort estimation [137].

Expert judgment differs from other estimation methods because the quantification step for generating the estimate is judgmental rather than mechanical [36]. That is, experts use their human mind as a measurement instrument [84]. Therefore, the processes that we use for arriving at a prediction are largely unconscious [37]. Discovering and understanding the factors that affect expert judgment estimates is crucial for reducing errors and improving our accuracy when using such a method, and research on these factors is also a trend [137]. In addition, research and practice in other domains where evaluations and predictions rely on expert judgment have shown that countless triggers can drive variability in judgments, leading to bias, noise — and consequently, to error, unfairness, and losses [85]. For instance, in the seemingly exact science of forensic fingerprint analysis, where professionals have to decide whether fingerprints collected in crime scenes match exemplar fingerprints, researchers found that examiners can be misled by contextual information, such as eyewitness recognition [86]. This led forensic laboratories to change their practices, sequencing information to which examiners are exposed before they analyze fingerprints.

Likewise, getting a comprehensive perspective of the factors researched in software estimation so far can guide researchers willing to build on the existing body of knowledge, to propose and assess new practices that minimize error and enhance the software estimation process. In addition, it can also help practitioners willing to identify the factors relevant to their context, to identify the good practices to adopt. In this article, we provide such perspective of factors through a Systematic Literature Mapping (SLM) using the guidelines of Kitchenham, Budgen, and Brereton [92] and Petersen et al. [131].

We found 131 relevant articles in our SLM, reporting 235 different factors — a myriad of diverse elements that somehow influence estimation results using expert judgment. Most of them (166 factors) was reported in one article and are provided as part of our supplementary material [114]. Still, understanding the remaining 69 factors investigated in two or more articles is challenging. Therefore, we propose an instrument for researchers and practitioners to navigate the seas of factors affecting estimates: the SEXTAMT (Software Estimates of eXperts: A Map of influencing facTors).

Typically, a sextant is an instrument to aid overseas navigation by measuring the angle between the horizon and a celestial reference object like the sun, planets, or stars. The celestial object chosen as a reference depends on the period of the day the observer will take a sight. The observer can use the sun during the day or planets and stars during dawn or night. The measured angle serves as input for calculations that allow for identifying positions with the aid of nautical charts, thus supporting navigation overseas. The time the observer took the sight is also a necessary input [46].

Likewise, the SEXTAMT uses reference points in the form of dimensions, which the interested reader can use to navigate these wide seas of factors. A temporal dimension alludes to the importance of time for calculating correct positions when using the physical sextant. In the SEXTAMT, it refers to a software project or iteration phases: initiating, planning, executing, monitoring and controlling, and closing — which we borrowed from the PMBOK (Project Management Body of Knowledge) group processes [163]. Most of the factors we found group at the planning and the executing phases. That is understandable because estimates emerge primarily at the planning phase, and the dynamics of project execution also affect our perceptions of accuracy and error of estimates.

Instead of finding a celestial object as a reference point, we included a stakeholder dimension to the SEXTAMT. The reader can define a stakeholder of interest to investigate only the factors associated with them, either because it relates to a task that the stakeholder is responsible for or because that stakeholder directly causes the factor. In some situations, the factor impacts the stakeholder somehow. Most factors are related to the estimator role, which is natural since stakeholders playing this role are responsible for estimating. However, we found factors associated with clients and users, higher management, project managers, requirement engineers, software developers, and testers. We also discovered factors that applied to the entire software team or no specific stakeholder at all.

The SEXTAMT also has a dimension regarding the type of effect of the factors. According to the direction of the effect, we had four types: positive direction for accuracy factors, negative direction for error factors, and neutral direction for value adjusting characteristics and empirical influence factors. If the reader wants to identify only the factors that increase accuracy when present, they can navigate the accuracy factors. Additionally, we grouped the factors in categories that represent the larger oceans and some smaller seas of our map.

2. Background

In this section, we present the relevant concepts for the context of our study (Section 2.1) and the related work Section (2.2), including two previous related reviews we found.

2.1. Software estimation

Software estimates are predictions about a variable, like the software project effort, cost, or duration [115]. Given the importance of software estimation for industry, one critical concern is to devise improved methods to estimate software projects. More than 60% of research papers about estimation before 2007 proposed and evaluated estimation methods [79]. Boehm classifies these methods as algorithmic models, expert judgment, analogy-based, Parkinson, price-to-win, top-down, and bottom-up [14]. Our SLM focuses on expert judgment estimation, as it is the most used method in the industry [149]. To delineate what we mean by expert judgment-method, we used the guideline of Halkjelsvik and Jørgensen [36]: if the quantification step of the estimation method is judgmental, then the method is categorized as expert-judgment-based. If this step is mechanical, then the method is categorized as model-based.

Another critical concern of software project estimation is the predicted variable, either size, effort, schedule, or cost of features [116]. For instance, the functions of algorithmic models use size as their input [60]. Then, considering software size estimates and productivity assumptions, estimators can generate effort estimates. From effort estimates and the project resources, estimators can generate estimates about cost, features, and duration (in calendar days), which project stakeholders use to establish the project commitments [116].

Nevertheless, many of the relationships among these software project variables are unstable and change from one context to another [65], hampering the creation of a universal model of estimation. This instability may also explain why complex estimation models are not necessarily more accurate than simpler ones [65]. Despite this, many of the existing estimation methods can be applied to any software project variables [117]. Therefore, in our SLM, we are not excluding studies based on the project variable.

2.2. Related Work

Researchers have been investigating factors affecting estimates such as the anchoring bias [5], the impact of the development method [124], the influence of using checklists [154], and others. In one of the related works, Halkjelsvik and Jørgensen [36] present a review of studies about factors affecting judgment-based predictions of performance time, integrating results from the areas of psychology, engineering, and management science. Their review later inspired writing a more recent book about time predictions in general [38]. Given the multidisciplinary nature of their review, they opted to term performance time predictions as an equivalent for effort estimation. The authors described (i) the characteristics of estimates presented in the primary studies (ii) the details about the processes and strategies used in estimation, and (iii) the influence of task characteristics, estimators' characteristics, and contextual factors on estimates.

Halkjelsvik and Jørgensen [36] included in their review studies correlational, quasi-experimental, and experimental designs. They excluded studies based on questionnaires and interviews describing respondents' opinions about reasons for estimation errors and biases because the authors affirm that they do not have a suitable method to evaluate their validity. Also, they have included gray literature, like reports and unpublished manuscripts, bringing back the practice perspective and the practitioners' voice to their results that otherwise would be lost because of the exclusion of studies based on questionnaires and interviews.

In another related work, Basten and Sunyaev [10] conducted an SLM focused on factors affecting software effort estimation accuracy. The authors presented four categories of factors affecting estimates: (i) factors related to the estimation process, (ii) factors related to the estimators' characteristics, (iii) features of the project to be estimated that may affect the estimates, and (iv) factors related to the external context, more specifically associated with the client. Although Basten and Sunyaev [10] published their SLM in 2014, they only included papers written up to 2010. Also, their search strategy consisted of a manual search and snowballing procedures [10]. An automatic search may provide additional papers. Diverging from Halkjelsvik and Jørgensen [36], Basten and Sunyaev [10] included papers reporting opinions from software experts, as they may indicate potentially influential factors.

Thus, we foresaw a need for an update and an expansion of such reviews. We executed our SLM on the scope of software engineering, including articles up to 2020, to satisfy this. On the one hand, our SLM differentiates from the review from Halkjelsvik and Jørgensen [36] by applying a systematic mapping method and focusing on the software engineering domain alone. On the other hand, our SLM differentiates from the review of Basten and Sunyaev [10] by extending the timeline of included papers, focusing on expert judgment only, and by including automated search instead of manual.

3. Research method

We started the SLM by defining a systematic mapping protocol, following the guidelines presented by Kitchenham, Budgen, and Brereton [92] and Petersen et al. [131], and by collectively inspecting it. The remaining of this section presents our research questions. It also presents our search, selection, extraction, and analysis procedures.

3.1. Research questions

Our primary research question is: ***RQ 1 – How have researchers investigated the factors that affect expert judgment software estimation?*** As we want to explore different aspects of the existing evidence about the factors, we further refined our primary research question in the following set of secondary research questions:

- **SQ 1.1 – What are the factors that affect expert judgment software estimation?**
- **SQ 1.2 – How was the impact of the factors over the expert judgment estimates measured?**
- **SQ 1.3 – What are the software project estimate variables investigated?**
- **SQ 1.4 – When and where are published the studies about factors affecting expert judgment software estimates?** and
- **SQ 1.5 – What research strategies and methods are used to investigate factors that affect expert judgment software estimation?**

3.2. Search and selection

We started the search process by defining a known set of papers, which we used as an oracle to validate our search string's outcomes. Our oracle had 25 papers[2]. Our next step was defining the search string. The results of automated searches are highly dependent on the search string's quality [131,162]. We defined ours based on the extraction of the keywords of the titles and abstracts from the articles in our known set of papers, as Petersen et al. [131] recommend.

We executed the automated search restricting the search to title, abstract, and keywords whenever possible. Our sensitivity[3] goal for the automated search was 70%, as Zhang et al. [162] recommended. After the first search round, we got a sensitivity of 60%—below our goal of 70%. We ran a trial search without restricting the search to title, abstract, and keywords, but the high number of results made this change prohibitive[4]. We refined the search string, leading us to the second and final version, presented in Table I.

Table I - Second version of the search string

("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 (factor OR reason OR cause OR "anchor" OR "impact" OR "risk identification" OR "customer collaboration") AND (software OR system)

[2] The final list with the known set of papers is in the supplementary material, together with more details of the search and selection procedures [114].
[3] Number of relevant studies retrieved divided by the total number of relevant studies and then multiplied by 100 [162]. The number of papers in the known set is the number of relevant studies.
[4] For ACM alone we had over 480,000 results.

We carried out the automated search on ACM, IEEExplore, Scopus, and EI Compendex (Engineering Village), as illustrated in Figure 1 (Step 1), resulting in 5,113 articles and a sensitivity of 84%, satisfying our goal of more than 70%. We did not include other publisher-specific databases, like SpringerLink and ScienceDirect, as they would probably yield a larger number of duplicates, according to Dyba et al. [23].

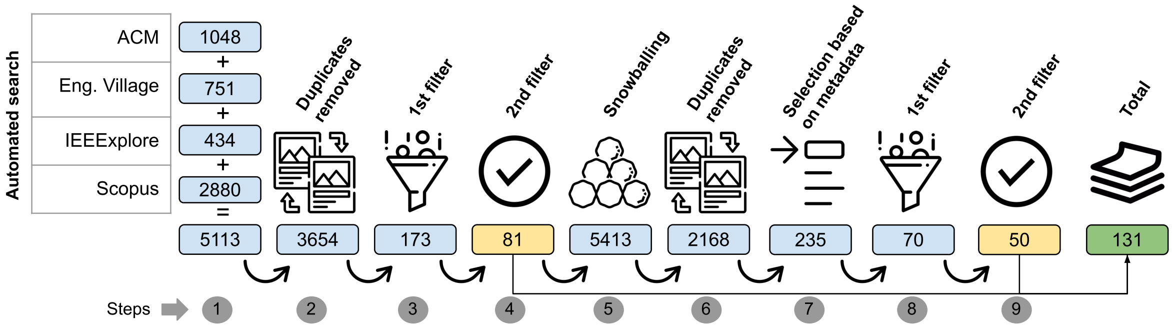


Figure 1 - Search and selection results

After eliminating duplicates from the 5,113 articles, we came to a total of 3,654 articles (Figure 1, Step 2). Next, we executed the selection procedures, considering the following inclusion criteria: IC01 – The paper presents an empirical study that investigates factors that affect software project estimates related to expert judgment. We also selected the papers based on the exclusion criteria that we present in Table II. Additionally, Table II presents the relationship between each exclusion criteria and the filter in which we applied it mostly: Filter 1 (title and abstract) and/or Filter 2 (full-text).

Table II - Exclusion criteria and their relationship with the selection filters.

ID	Exclusion criteria description	Filter
EC01	The paper presents a systematic mapping/review, lessons learned, or opinion paper, rather than an empirical study on factors that affect software project estimates related to expert judgment.	1, 2
EC02	The paper focus on factors affecting estimates related to estimation methods other than expert judgment.	1, 2
EC03	The paper presents non-peer-reviewed results.	1
EC04	The paper is not written in English.	1
EC05	The paper is not accessible in full-text online.	1
EC06	The study is published as a book or grey literature.	1
EC07	The paper is a duplicate or a previous version of another already selected paper.	2
EC08	The paper does not describe the factors to allow for categorization	2

To reduce bias during the selection process, we independently selected a random sample of the articles retrieved by the search by reading their titles and abstracts. We calculated the researchers' level of inter-rater agreement on this sample of articles through the kappa coefficient [92]. We got a kappa level of 0.83, which is very good, according to Kitchenham et al. [92]. So, we considered the kappa level adequate, and we proceeded with the selection, getting to a total of 173 papers selected based on title and abstract (Figure 1, Step 3). After reading the full text of all the 173 articles, we selected 81 that satisfied the inclusion criteria and that we could not eliminate with our exclusion criteria (Figure 1, Step 4).

The final set of papers selected from the database search formed the start-set for backward and forward snowballing [158]. We aimed for a sensitivity of 100% after the snowballing step. We got to a total of 5,413 articles through backward and forward snowballing (Figure 1, Step 5), and to 2,618 after removing duplicates (Figure 1, Step 6). We selected a total of 234 of them based on their metadata - title, authors, and venue - and on their citation context on the original articles in the case of backward snowballing (Figure 1, Step 7). We read their abstracts, reducing the number to a total of 70 articles (Figure 1, Step 8). Following, we read their full text, leading to the inclusion of 50 articles (Figure 1, Step 9). Therefore, the final list of articles included in our SLM contains 131 articles, and we satisfied our goal of 100% sensitivity of papers from our known set of papers.

3.3. Data extraction

We extracted the data using a form[5] created and later refined after a pilot data extraction over the known set of papers. We extracted the following data:

- Title, authors and their affiliation, venue and year of publication;
- research strategy, according to the classifications of Stol and Fitzgerald [141] and Storey et al. [142], and research method;
- observations and context;
- factors and discussion about them;
- project variables that were the focus of estimation. These variables could be either size, effort, cost, productivity, or duration;
- how authors measured the impact of the factors over the estimates.

3.4. Data analysis

In Figure 2, we provide an overview of our data analysis. After reading the full text of all selected articles and extracting text and data to our extraction form, we created codes to

[5] The form, as well as the complete extraction data are in the supplementary material [114].

summarize the findings from the primary studies[6], supporting the aggregation of data into factors later during the analysis process.

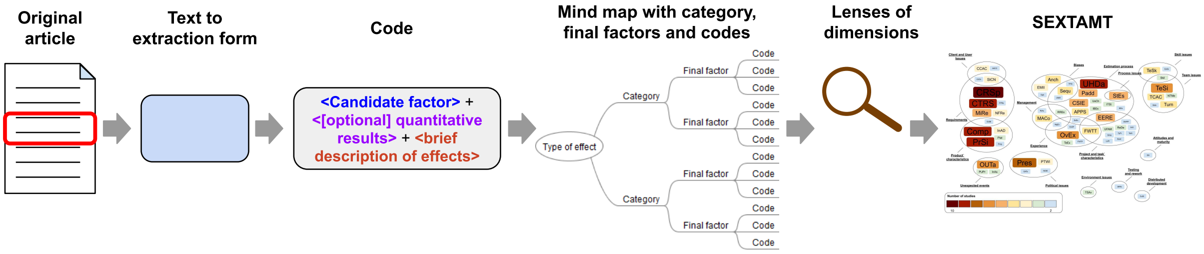


Figure 2 - Overview of the analysis

Most of the codes we generated followed the structure we show in Figure 2, with some variations. The **candidate factor** was the label that the original study authors provided. The **quantitative results** summarized whether the authors found significant results, sometimes informing p-values or other relevant information. It was optional, once only quantitative studies needed such data. The **brief description of effects** highlighted whether the candidate factor was a reason for accuracy, a reason for errors, an effort predictor, among others.

Next, we created mind maps aggregating similar candidate factors under a final factor label. We chose the final label to reflect the core of the candidate factors. In some situations, we had an intermediary factor label, reflecting essential variations of the core factor. We held regular meetings to review the mind maps with the categories, candidate factors, and codes. We analyzed the factors through the lenses of a few dimensions we considered relevant to interpret the results. The categories we used to organize the data relate to three dimensions, shown in Figure *3*.

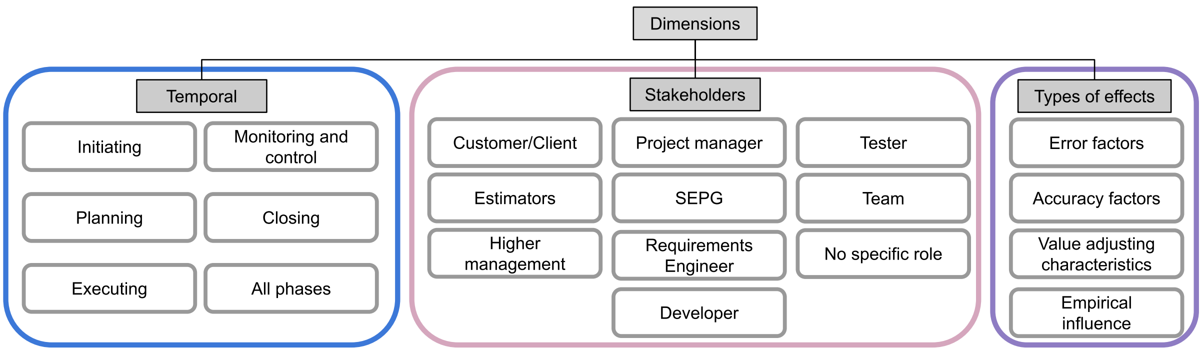


Figure 3 - SEXTAMT dimensions

The temporal dimension regards the phase of a software project/iteration that a factor is likely to happen or to cause an impact, based on the PMBOK project phases [163]. The stakeholder dimension informs one stakeholder or a group responsible for a task or process

---

[6] All factors with their categories and codes are in the supplementary material [114].

to which the factor is linked or that directly causes the factor. In some situations, the factor impacts the stakeholder. The type of effect dimension indicates the nature of the impact of the factor over the estimates, considering the results of the primary studies: (i) error factors are negative when present; (ii) accuracy factors lead to improvements in estimates' accuracy when present; (iii) value adjusting characteristics lead to a need for a higher or lower value of estimate and are inputs to estimation; and (iv) empirical influence indicate factors whose impact on the estimates are not definitely negative, positive, or leading to a need to a higher or lower value: it varies in direction and nature. Some of the factors under this label can lead to improvements in accuracy in some circumstances, but to inaccuracies in others. For instance, the client's expectation factor has an empirical influence over the estimates. If, by chance, such expectations are realistic, their impact are on the direction of making the estimate more accurate. Otherwise, they may lead to estimation error.

Finally, we created the SEXTAMT. We used the dimensions as the cornerstone for the navigation through the factors. However, we excluded from the SEXTAMT all the factors reported in only one article due to space restrictions, reporting them in our supplementary material. In the next section, we explore our results.

4. Results

In this SLM, we aim to answer the following primary research question: ***RQ 1 - How have researchers investigated the factors that affect expert judgment software estimation?*** In this section, we explore our results, considering each secondary research question presented in Section 3.1.

4.1. SQ 1.1 – What are the factors that affect expert judgment software estimation?

After analyzing all papers, we found 235 factors in total, from which we report the 69 that were explored in more than one research article. We present the 69 factors in *Table* III, with an ID code in parenthesis, and the articles with the evidence about them.

Table III - List of factors

Factor	Articles
Diligence (Dili)	[9] [99]
Anchoring effect (Anch)	[138] [5] [105] [56] [73]
Effect of more and/or irrelevant information (EMII)	[72] [152] [55] [30]
Optimism (Opti)	[71] [109]
Sequence effects (Sequ)	[31] [69] [75] [50]
Time frame size (TFSi)	[74] [35]
Unit effects (UnEf)	[69] [47]
Size (PrSi)	[21] [153] [152] [94] [95] [140] [41] [157] [151]

Factor	Articles
Complexity (Comp)	[21] [154] [108] [127] [101] [140] [160] [143] [1]
Integration and dependencies (InAD)	[17] [108] [153] [101]
Platform (Plat)	[94] [45] [1]
Programming language (Prog)	[41] [45]
Collaboration and communication (CCAC)	[99] [151] [100] [123]
Availability of knowledgeable/competent clients (AKCC)	[113] [32]
Client’s expectations (ClEx)	[72] [81]
Clarity of client's needs (ClCN)	[99] [113]
Cultural diferences (CuDi)	[17] [1]
Tool support and avaliability (TSAv)	[101] [57]
Use of historical data (UHDa)	[159] [110] [99] [21] [101] [24] [57] [133] [139] [100]
Padding (Padd)	[110] [99] [57] [100] [26] [97]
Anticipation of project' participants' skills (APPS)	[159] [132] [99] [152] [100]
Use of checklists (UsCh)	[154] [24] [57]
Combination strategy of individual estimates (CSIE)	[77] [111] [25] [40] [126] [125]
Involvement of technical staff (ITSt)	[159] [99] [3]
Informal basis for estimating (IBEs)	[153] [96] [91]
Impact of early estimates (IEEs)	[70] [80]
Reestimation and revision of estimates (REEs)	[152] [94] [95]
Standards in estimation (StEs)	[159] [110] [99] [133] [3] [100]
Enough effort and resources spent on estimation (EERE)	[159] [70] [99] [57] [73] [133]
Overall experience (OvEx)	[17] [153] [127] [109] [113] [57] [90]
Technical experience (TeEx)	[21] [39] [1]
Experience with similar/previous projects/tasks (ExSP)	[57] [151]
Familiarity with the product (FWTP)	[101] [22]
Estimation experience (EsEx)	[133] [1]
Manager experience (MgEx)	[127] [2]
Monitoring and control (MACo)	[159] [127] [57] [32] [91]
Risk assessment (RiAs)	[159] [127]
Pressure (Press)	[159] [99] [109] [161] [91] [26]
Price-to-win issues (PTWI)	[159] [153] [109] [57]
Goals and targets (GATa)	[110] [109]
Negotiations games in estimates (NGIE)	[109] [26]
Use of flexible/agile development model (UFAM)	[124] [93] [18]
Resources dependencies (ReDe)	[21] [152] [95]
Simplicity (Simp)	[57] [73]
Project flexibility (PrFl)	[57] [32]
Similarity with previous tasks/projects (SWPP)	[57] [73]
Task size (TaSi)	[154] [43]
Business area (BuAr)	[41] [45]
Type of project (TyPr)	[41] [1]
Longer projects (LoPr)	[94] [41]
Familiarity with the technology (FWTT)	[9] [101] [24] [73] [91]
Clear requirements specification (CRSp)	[159] [99] [21] [153] [24] [57] [32] [2] [161] [7]
Changes to requirements or scope (CTRS)	[159] [153] [152] [95] [32] [160] [7] [91] [100] [99] [113] [57]

Factor	Articles
Misunderstanding of requirements (MiRe)	[21] [109] [113] [57] [73] [91]
Non-functional requirements (NFRe)	[153] [101] [140] [151]
Familiar problem or requirements (FPRe)	[95] [73]
Dependencies between user stories/backlog items (DUBI)	[21] [1]
Technical skill (TeSk)	[71] [24] [57] [53] [91]
Estimation skills (EsSk)	[110] [91]
Training in Estimation (TrEs)	[159] [133]
Team Size (TeSi)	[21] [94] [140] [41] [45] [1] [43]
Team Collaboration and communication (TCAC)	[159] [17] [152] [113] [1]
Turnover (Turn)	[99] [102] [108] [153] [103]
New team members (NTMe)	[159] [21] [91]
Team Stability (Stab)	[153] [140]
Team Skill (Skil)	[17] [153] [151]
Overlooked and unplanned tasks (OUTa)	[99] [21] [24] [109] [57] [73] [100]
Incorrect assumptions (InAs)	[21] [24] [73]
Occurrence of unforeseen problems (OUPr)	[159] [21] [73]

In Section 5, we detail the factors, presenting them as part of the SEXTAMT. We also organized the factors considering the dimensions we described in Figure *3*.

4.2. SQ 1.2 – How was the impact of the factors over the expert judgment estimates measured?

This question's motivation was to identify how researchers evaluate the impact of the factors over the estimates. Table IV presents the associations between the strategy that researchers used for impact measurement with each article. Each article could have multiple different ways to measure impact.

Table IV - Impact measurement strategy by article.

Impact measurement strategy	Article
Difference of estimates	[31] [70] [130] [138] [5] [72] [71] [155] [63] [122] [76] [62] [94] [80] [29] [105] [8] [61] [56] [150] [74] [81] [55] [30] [118] [69] [18] [143] [45] [144] [139] [58] [53] [75] [47] [66] [35] [147] [78] [43] [126] [50] [74] [125]
Participants' perception	[159] [132] [17] [70] [130] [110] [82] [99] [21] [102] [154] [108] [153] [148] [152] [42] [145] [101] [80] [109] [120] [140] [98] [113] [57] [32] [73] [133] [134] [136] [2] [151] [3] [160] [1] [107] [135] [161] [7] [68] [103] [96] [100] [26] [97]
MRE	[124] [9] [71] [33] [120] [57] [156] [73] [90] [25] [34] [40] [44] [20]
MREBias	[124] [57] [73] [64] [40]
BRE	[124] [152] [111] [27] [123] [125]
BREBias	[124] [9] [154] [152] [24] [111] [27] [125]
Deviation	[112] [70] [12] [119] [39] [103] [129]
Absolute error	[130] [25] [53]

Impact measurement strategy	Article
Total effort	[120] [41] [157] [6] [128] [22]
Interval of over/underrun (over/underestimation)	[17] [99] [11]
Pred(X)	[9] [28]
Confidence related	[82] [77] [105] [49] [30] [73] [69] [52] [34] [83]
Not informed/not defined	[13] [15] [19] [127] [93] [91] [48] [146] [16]
Other	[130] [51] [95] [54] [104] [156] [67]

Researchers' most used strategy for investigating the impact of factors was participants' perceptions: 45 articles adopted it, using either respondents or field research strategies. Some of these studies required participants to evaluate their companies or project accuracy subjectively. Another strategy widely used was assessing the difference of estimates between an experimental and a control group, with 44 occurrences. This is common in laboratory experiments, which was the most applied research strategy discussed in Section 4.5. By analyzing the difference of estimates, researchers investigated the factors that could cause a shift from more realistic estimates to more optimistic ones — supposing that lower estimates lead to higher chances of error. Regarding more objective measures of accuracy, bias, and error, researchers used metrics like MRE (Magnitude of Relative Error), MREBias, BRE (Balanced Relative Error), and BREBias, as we show in Table V.

Table V - Objective metrics of accuracy, bias, and error.

Accuracy Metrics	#	Bias Metrics	#
MRE	13	MREBias	5
BRE	6	BREBias	8

Seven studies relied on less traditional metrics involving the estimated and actual values. While the critiques of MRE and MREBias focus on the use of actual values at the denominator of the formula — which is resolved in BRE and BREBias by using the minimum value between the estimated and actual values — seven studies use the estimated value at the denominator. We categorized these studies under the term "deviation", since the researchers of such articles disagree about the best name for the metric, calling it effort deviation [112,129], effort overrun [70], accuracy [12], effort variance [119], overrun factor [39][7], or project overrun [103]. Another three studies use the absolute error (estimated - actual value).

[7] The original formula was actual duration = estimated value + estimated value*overrun factor for this study. Isolating the overrun factor, we get to the same formula as the other studies.

A total of six studies evaluates total effort. They are either based on regression analysis ([41,120]) or correlations of effort with other variables ([6,22,128,157]). Three studies relied on classifying projects according to ranges of over/underestimation or over/underruns. Two of them were respondent studies, and therefore the classification depended on respondents' memories ([17,99]). The other study was a data one ([11]). Also, two studies used pred(x) [59].

4.3. SQ 1.3 – What are the software project estimate variables investigated?

Regarding the project variables investigated in the primary studies, we extracted the metrics that authors reported as within their studies' scope. Figure 4 shows the results we obtained, making evident that most of the studies focus on effort estimation.

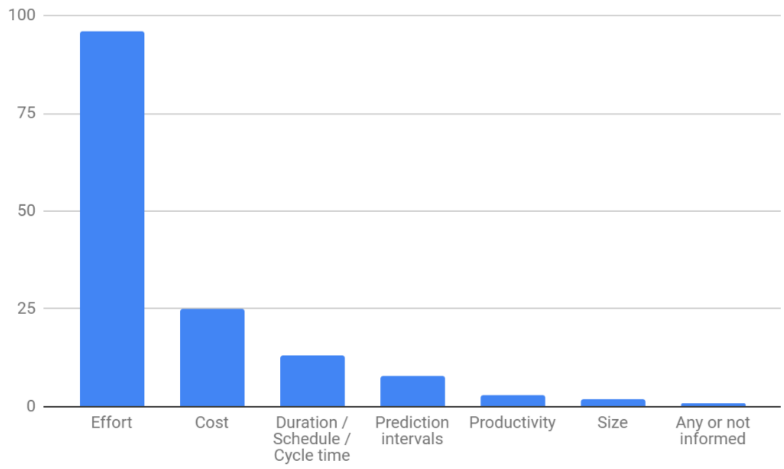


Figure 4 - Variables investigated in primary studies

Most of the studies focused on effort estimation (96 in total). Twenty-five studies claimed to investigate factors related to cost, while 13 focused on duration. Eight studies explored prediction intervals — mostly of effort — and we classified them separately to emphasize the importance of avoiding single values when estimating. Three studies reported factors associated with productivity. Only two studies claim to investigate factors associated with size, probably because the focus is on other metrics when using expert judgment.

4.4. SQ 1.4 – When and where are published the studies about factors affecting expert judgment software estimates?

Our sample includes articles published between 1989 and 2020. The past two decades have been very fruitful regarding research about factors affecting estimates, as shown in Figure 5, revealing an increasing interest in them. We also show a trendline reporting the moving average (past five years), revealing a relative degree of stability of the number of papers published regarding factors affecting expert judgment estimates since 2016.

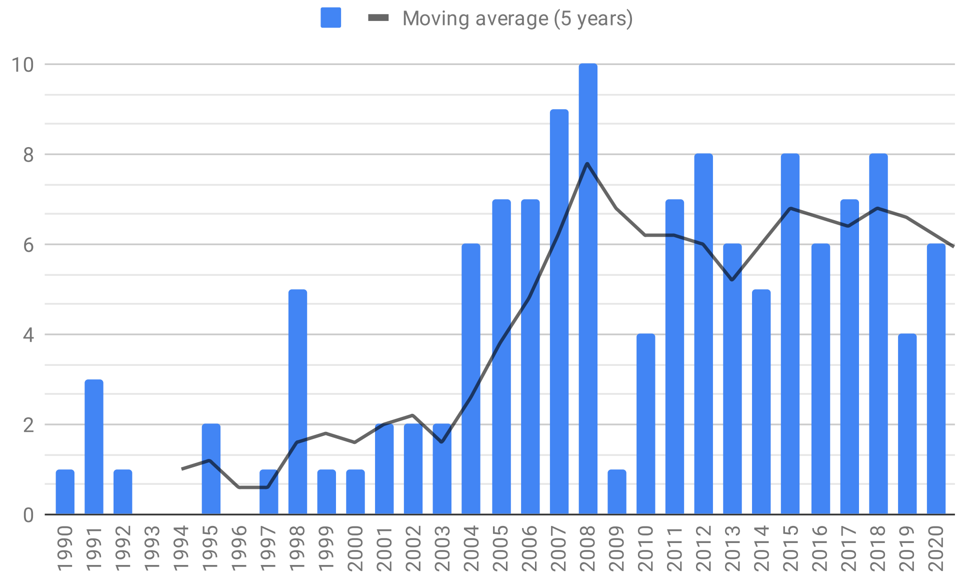


Figure 5 - Research about factors affecting the estimates over the years

Error! Reference source not found. shows all the venues concentrating three or more studies about factors affecting estimates. In total, we represent 65 articles in Table VI.

Table VI - Top venues

Venue	# citations
Journal of Systems and Software	15
IEEE Transactions on Software Engineering	10
Information and Software Technology	8
Euromicro Conference on Software Engineering and Advanced Applications	5
International Conference on Evaluation and Assessment in Software Engineering	5
IEEE Software	4
Empirical Software Engineering	4
International Symposium on Empirical Software Engineering and Measurement	4
International Conference on Product Focused Software Process Improvement	4
International Journal of Project Management	3
International Software Metrics Symposium	3

There is a balance between publishing in conferences (63 occurrences) and journals (68 occurrences). The Journal of Systems and Software, IEEE Transactions on Software

Engineering and Information and Software Technology, concentrated the highest number of articles.

4.5. SQ 1.5 – What research strategies and methods are used to investigate factors that affect expert judgment software estimates?

To answer SQ 1.5, we classified the studies considering the taxonomies proposed by Storey et al. [142], which is focused on human factors of software engineering, identifying four research strategies: respondents, lab, field, and data, as we show in *Table* VII. Each article can report more than one study and, accordingly, could be associated with more than one research strategy.

Table VII - Research strategies distribution

Research strategy	Number of studies
Data	31
Field	31
Lab	51
Respondents	33

In general, the different available research strategies had been used in a balanced way, except for lab strategies, which detach from the others as the most used one. That is, most of the studies in our sample evaluate one factor in a controlled setting through hypothesis testing [142]. Studies investigating or reporting more than one factor generally employ respondent or field strategies, each one having 33 and 31 occurrences, respectively, in our data. In Figure 6 we show the use of the research strategies throughout the years.

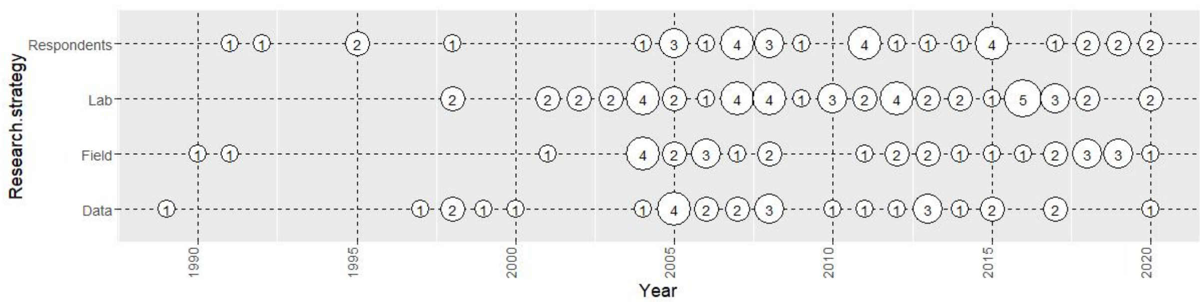


Figure 6 - Research strategies throughout the years

Research about factors affecting estimates became prolific after the year 2005. Since then, the distribution of studies using different strategies has been relatively uniform. However, it seems that laboratory strategies are outperforming the others in the past decade.

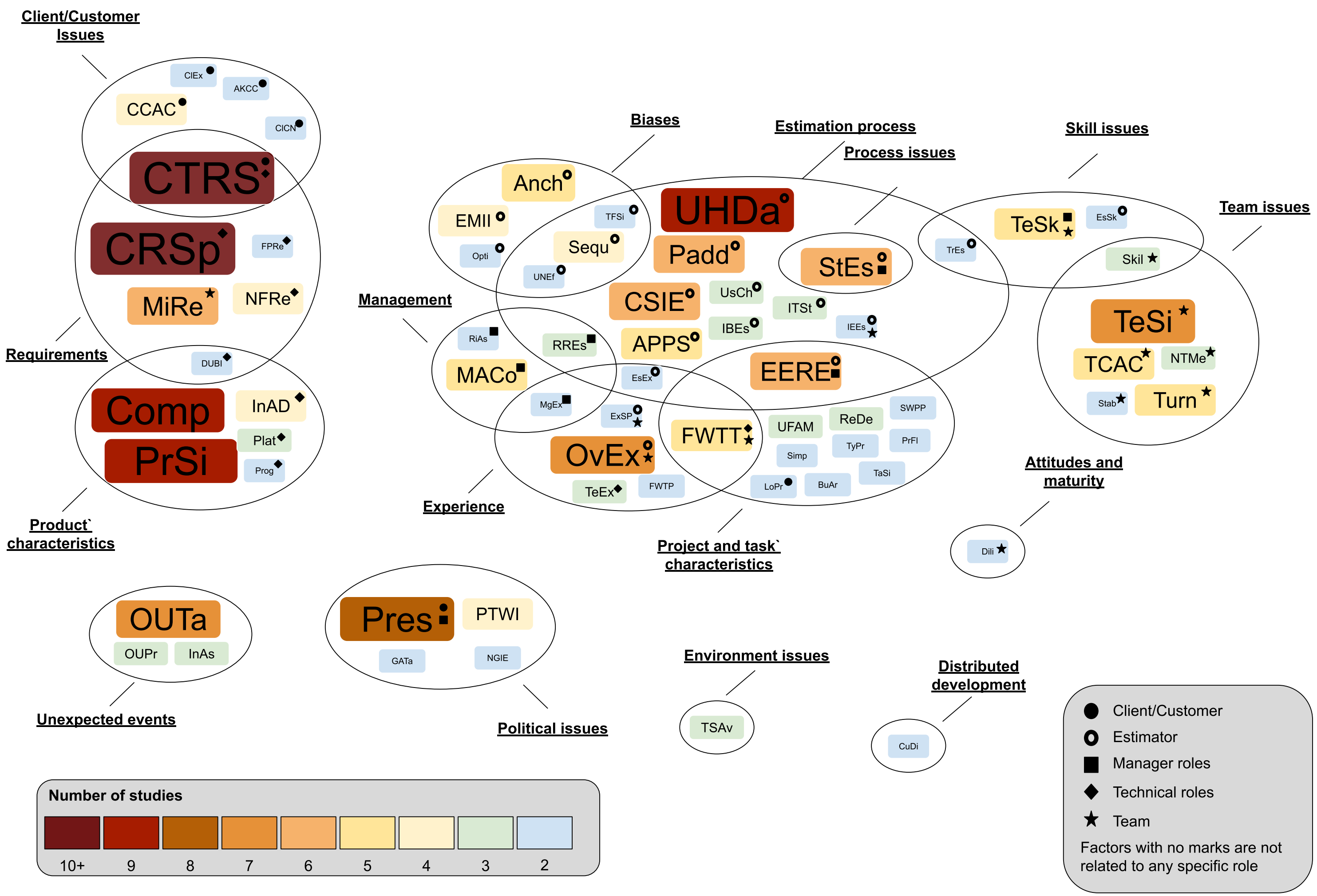
5. The SEXTAMT

As we informed in Section 4.1, we found a total of 235 factors, of which 69 were reported in two or more articles. We gathered these 69 factors in one instrument: the SEXTAMT. It has three dimensions to allow the navigation through the seas of factors:

1. The temporal dimension provides a view of the factors relevant for different software project or iteration phases.
2. The stakeholders' dimension focuses on the factors associated with different roles in the software process.
3. The type of effect' dimension, based on the direction of the effect of the factor.

In Figure 7, we present the overall map of factors affecting estimates — a bird's eye view of the SEXTAMT. We represent the factors as rounded rectangles, labeled with the factors' codes we indicated in **Error! Reference source not found.***Table* III. We marked some of them with symbols related to their stakeholders' dimension. The size and color of each factor represent the number of articles investigating it. We also grouped them by major categories represented in the form of ellipses. We also provided an expanded view of Figure 7 as part of our supplementary material, in which we added the studies that investigated each factor.

AKCC	Availability of knowledgeable/competent clients
Anch	Anchoring effect
APPS	Antecipation of project' participants' skills
BuAr	Business area
CCAC	Collaboration and communication
ClCN	Clarity of client's needs
ClEx	Client's expectations
Comp	Complexity
CRSp	Clear requirements specification
CSIE	Combination strategy of individual estimates
CTRS	Changes to requirements or scope
CuDi	Cultural differences
Dili	Diligence
DUBI	Dependencies between user stories/backlog items
EERE	Enough effort and resources spent on estimation
EMII	The effect of more and/or irrelevant information
EsEx	Estimation experience
EsSk	Estimation skills
ExSP	Experience with similar/previous projects/tasks
FPRe	Familiar problem or requirements
FWTP	Familiarity with the product
FWTT	Familiarity with the technology
GATa	Goals and targets
IBEs	Informal basis for estimating
IEEs	Impact of early estimates
InAD	Integration and dependencies
InAs	Incorrect assumptions
ITSt	Involvement of technical staff
LoPr	Longer projects
MACo	Monitoring and control
MgEx	Manager experience



MiRe	Misunderstanding of requirements
NFRe	Non-functional requirements
NGIE	Negotiations games in estimates
NTMe	New team members
Opti	Optimism
OUPr	Occurrence of unforeseen problems
OUTa	Overlooked and unplanned tasks
OvEx	Overall experience
Padd	Padding
Plat	Platform
Pres	Pressure
PrFl	Project flexibility
Prog	Programming language
PrSi	Size
PTWI	Price-to-win issues
ReDe	Resources dependencies
RiAs	Risk assessment
RREs	Reestimation and revision of estimates
Sequ	Sequence effects
Simp	Simplicity
Skil	Skill
Stab	Stability
StEs	Standards in estimation
SWPP	Similarity with previous tasks/projects
TaSi	Task size
TCAC	Collaboration and communication
TeEx	Technical experience
TeSi	Size
TeSk	Technical skill
TFSi	Time frame size
TrEs	Training in estimation
TSAv	Tool support and avaliability
Turn	Turnover
TyPr	Type of project
UFAM	Use of flexible/agile development model
UHDa	Use of historical data
UnEf	Unit effects (Lower granularity time unit)
UsCh	Use of checklists

Figure 7 - The SEXTAMT

Figure 7 shows two larger oceans, formed by categories that share common factors. The larger one contains the categories: estimation process, biases, management, experience, skill issues, team issues and project and task characteristics. It also concentrates many of the top investigated factors: the *use of historical data*, *padding*, the *combination strategy of individual estimates*, *standards in estimation*, *enough effort and resources spent on estimation*, *overall experience*, and *team size*.

Client/customer issues, requirements, and product' characteristics are categories that also share factors, forming another larger ocean with some of the factors that stand out: *changes to requirements or scope*, *clear requirement specifications*, *misunderstanding of requirements*, *complexity*, and *product size*. The map also has some categories representing smaller seas, of which political issues and unexpected events are the larger ones. *Pressure* and *overlooked and unplanned tasks* are the most investigated factors, respectively.

The remaining of this section describes the factors composing the SEXTAMT in more detail, from the perspective of dimensions we presented in Figure *3*. In each of the following subsections, we show the factors for each different class of stakeholders, organizing them per project phase. Therefore, the reader may easily navigate through the factors by stakeholder and by phase. We also present the type of effect for each factor.

5.1. Customer/Client

Figure 8 shows all the factors related to customers/clients, each one represented by a blue box. We wrote the factors using positive statements representing the presence of a factor, like in *the clarity of the client's needs*, representing such presence through green circles in Figure 8. However, the existing evidence may refer to the absence of such an aspect, like *the lack of clarity of the client's needs*, represented in Figure 8 by a red circle inside the factor box. Figure 8 also presents the timeline of the typical project or iteration phases when a factor may happen or cause an impact over the estimates: the temporal dimension of the SEXTAMT. We also mapped each factor to their type of effect at the right of the figure. Some factors are organizational or overarching, and we represent them at the left of the image. We did not present their types of effects on the figure to keep it simple: we discuss it in the text only. In addition, the gray hexagons associated with each factor represent the articles that published results regarding them. The numbering of each hexagon indicates the article ID in the extraction forms (part of our supplementary material).

At the **planning phase**, four factors stand out. Two studies report findings related to *the lack of clarity of client's needs* as an error factor. Lederer and Prasad [99] present a survey where the users' lack of understanding of their requirements is a reason for inaccuracy. Matos et al. [113] report a qualitative study where clients who do not know what they want hinder software estimation and accuracy in the context of web effort estimation. Other two studies report that *longer projects* relate to higher costs [94] and that increasing calendar time will increase total effort [41]. Therefore, it is a value adjusting characteristic.

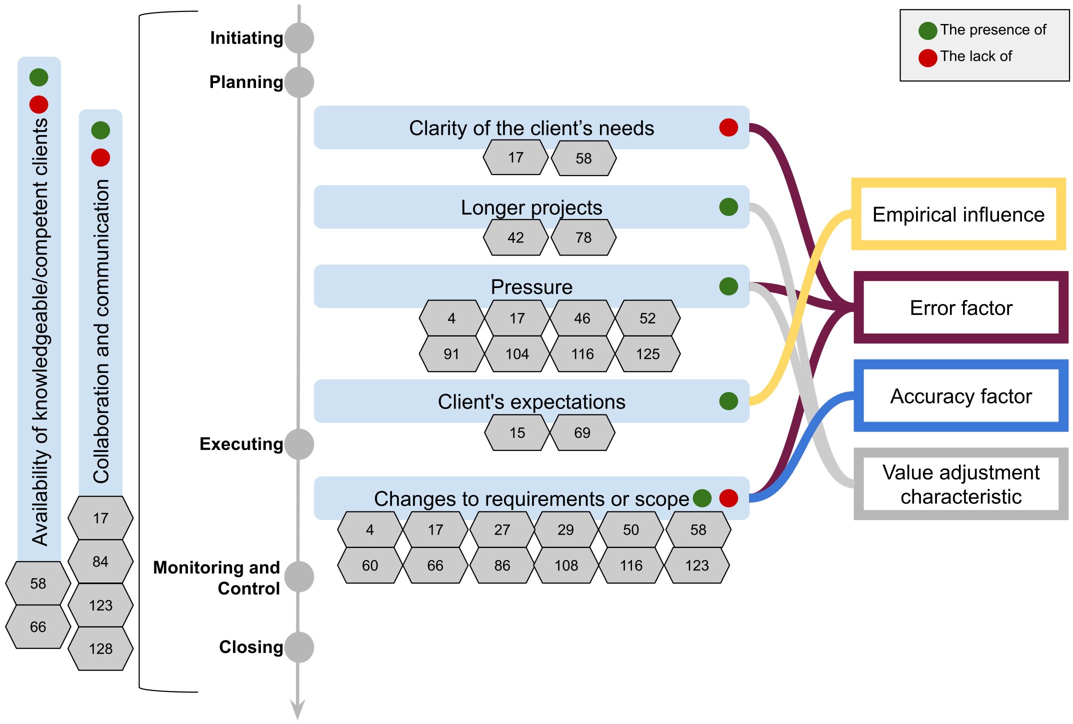


Figure 8 - Factors related to Customer/Client

Eight studies declare that *pressure* impacts estimating, either as an error factor or as a value adjusting characteristic. Nevertheless, the articles describe pressure in varying levels and originating from different sources. It can, for example, be an overall pressure, directed by management or related to the schedule alone. Therefore, we created intermediary factors for pressure, and in this section, we explore only the customer pressure, which appears in two studies. Yang et al. [159] point out that pressure from senior managers and clients to set or change the estimation results is a reason for inaccurate estimates. Keaveney and Conboy [91] report that pressures from customers or managers result in lower estimates than would be realistically expected.

The final factor at the planning phase is the *client's expectations*, which have an empirical influence over the estimates. Estimators were impacted by the effort informed by the client at the specification of one experiment [81]. This result repeated even when estimators are told to disregard such information [72].

At the **executing** phase, *changes to requirements or scope* emerge as an error factor when present, with twelve studies discussing it. Some studies report that requirement changes are a reason for inaccuracies [95,153,160], and two studies indicate that frequent changes are the problem [7,99]. Others emphasize that requirement changes contribute to overruns [32,39], are a challenge [152], or a potential problem for estimation [91,100]. Finally, some researchers identify changes in scope [95] and scope creep [57,153] as reasons for inaccuracies. When the client's needs are stable, it facilitates software estimation and raises accuracy [113], so the absence of changes to requirement or scope is an accuracy factor.

Some factors intersect **all phases**. For instance, the availability of clients who understand the project's business rules facilitates software estimation and accuracy [113] - therefore, *the availability of knowledgeable/competent clients* is an accuracy factor. Moreover, the lack of it leads to errors, as the client's unavailability hinders software estimation and accuracy [113], and the lack of competent customers able to make decisions is a reason contributing to overruns [32]. *Collaboration and communication* with the customer and users is an additional factor trespassing all phases. Researchers report that good collaboration with customers, facilitated by frequent communication, was associated with projects that experienced a lesser magnitude of effort overruns [123]. Also, researchers found that insufficient user-analyst communication and understanding was a potential cause of estimating problems in a case study [100], confirming it is a reason for inaccuracy later on in a survey [99]. Additionally, in the agile context, customer communication is an effort predictor [151]. Thus, *collaboration and communication* with the customer and users is an accuracy factor and a value adjusting characteristic. When absent, it is also an error factor.

5.2. Estimator

Figure 9 presents all the factors related to anyone assuming the role of an estimator. Only one factor is related only to the **initiating** phase: *early estimates* - two studies indicate that they impact estimates in later phases [70,80]. In one of them, project leaders believed that pre-planning estimates impacted detailed estimates, although they could not express the extent of the impact. In a laboratory experiment later, the researchers confirmed the existence of the effect [80]. In a field experiment about project bidding, companies providing early price indications based on limited and uncertain information gave higher estimates in the next bidding round. Such findings surprised the researchers, who expected the early estimates to act as anchors, leading to lower bids. Next, they carried out a laboratory experiment to explore further this finding, concluding that early estimates act as anchors to final estimates only when estimators have nothing to lose [70].

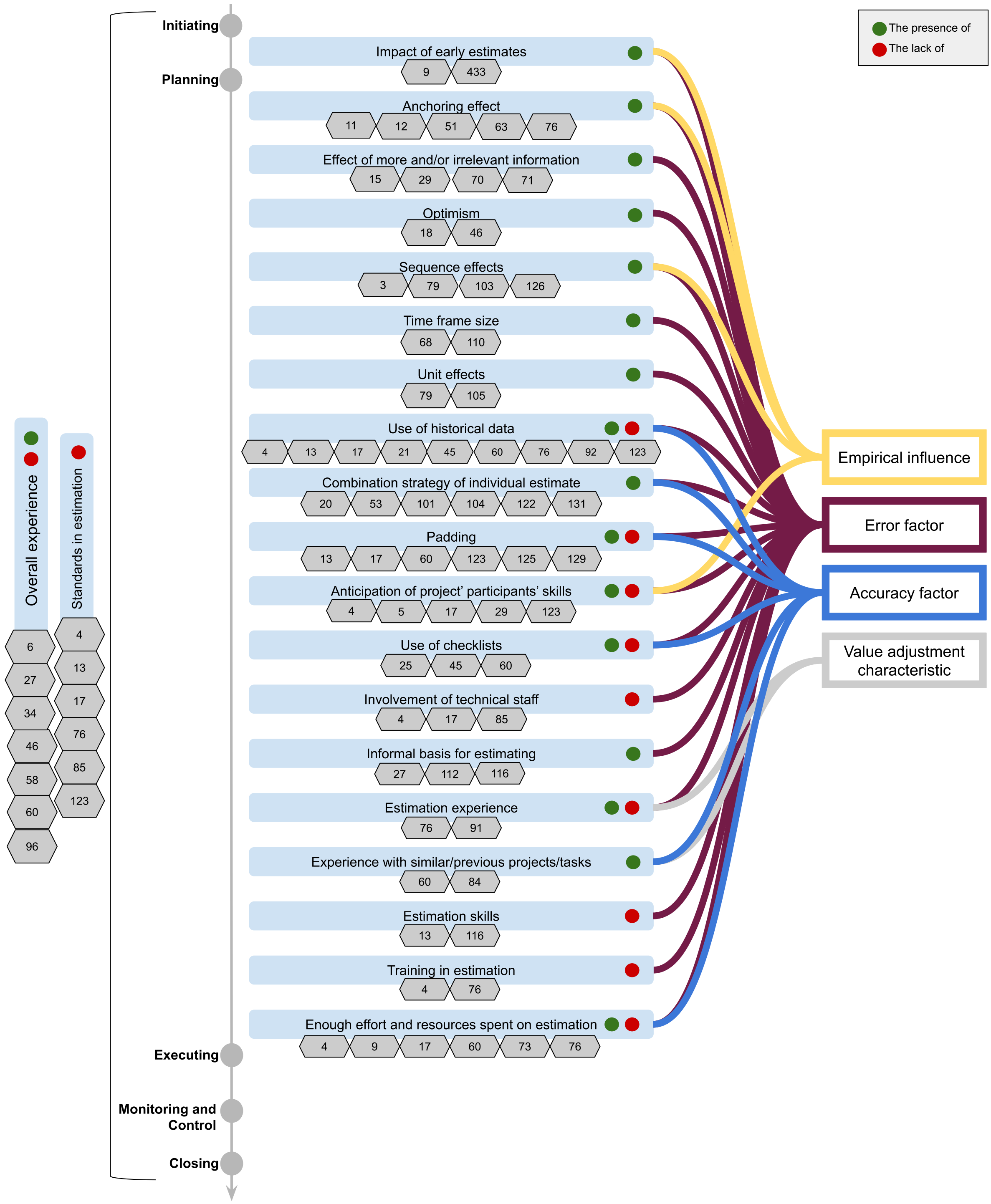


Figure 9 - Factors related to Estimators

All the other factors mapped to estimators concentrate on the **planning** phase. Many of them are biases, such as the *anchoring effect*, which is our tendency to be influenced by values presented to us before the estimation activity [105]. In a field study it is reported to hinder the creation of a meaningful estimate [133] and, thus, is an error factor. Many laboratory experiments also report that the anchoring effect impacts software estimation [5,56,105,138] — therefore providing evidence of its empirical influence over the estimates. Aranda and Easterbrook [5] found a statistically significant impact of numerical anchors on time estimates.

Jørgensen and Grimstad [56] also found a significant impact of numerical anchors over estimates, reporting a medium to large effect size. They also found a small to medium effect size when using a textual anchor: putting the same requirements specification as a “minor extension” work led to lower estimates than putting it as “new functionality” work. Løhre and Jørgensen [105] found a slight tendency for a larger anchoring effect with interval anchors compared to single value anchors when dealing with numerical anchors. Additionally, they expected the expertise — defined as the length of experience — of the anchor's source would act as a moderator for the anchoring effect. Surprisingly, they found that the receiver’s expertise that acted as such. Beyond investigating anchoring itself, Shepperd, Mair, and Jørgensen [138] discovered that raising awareness about anchoring reduces the impact of high anchors on productivity estimations but does not eliminate the effect.

Another relevant factor for estimators is the *effect of more and/or irrelevant information* over the estimates. Usman et al. [152] found that the availability of more detailed information may increase underestimation bias by increasing estimator’s optimism. Grimstad and Jørgensen [30] report that specifications with irrelevant information lead to higher estimates in laboratory experiments. Jørgensen and Grimstad [72] explored different aspects of irrelevant and misleading information that have an effect over the estimates: (i) the client's cost expectations, (ii) the wording of the specification (words associated with small and simple tasks lead to underestimation, while words associated with complex and large tasks lead to overestimation), (iii) the suggestion of future opportunities for work contingent on performance in current projects (lead to underestimation), and (iv) the amount of information, even when they are irrelevant (more information leads to overestimation). Asking people to highlight relevant information or strike irrelevant ones is not enough to eliminate the observed impact [72]. Additionally, in a field experiment, Jørgensen and Grimstad concluded that informing that the customer required development in a short period with start-up several months ahead also led to lower estimates, though supposedly this information is irrelevant to estimation [55].

Optimism is an additional error factor, leading to estimates' unintentional distortions, for instance [109]. Jørgensen, Faugli, and Gruschke [71] measured general optimism in varying ways in an experiment. They discovered that explanatory style, life orientation, and higher self-assessed level of optimism are all weakly connected with optimistic predictions. Also, merely asking estimators whether they assess themselves to be more or less optimistic seems to be enough as an indicator of optimistic predictions - instead of using more complex measures of optimism as the scales for explanatory style or life orientation [71].

Estimators should also be aware of *sequence effects* relative to the order of estimation of tasks and projects with different sizes[8]. Grimstad and Jørgensen [31] showed a statistically significant difference when starting estimation with a small task, compared with starting with a large one first. Jørgensen [50] also investigated the estimation of a large and a small system, with a significant result when inverting the reference task's order — that is, the one estimated first. When estimating projects of similar sizes in a sequence, estimators tend to estimate the target project as more extensive compared to the reference project [50]. Another set of experiments reverberated that for differently sized tasks the estimate is biased to become more similar to the one of the previous task in the sequence. In contrast, for similarly sized tasks, the estimation is biased to become more different than the previous one [75].

Two articles address the *time frame size*: shorter time frames tend to lead to more optimistic estimates than larger ones [35,74]. Another two articles investigate *unit effects*: asking for estimates using a lower granularity time unit led to lower estimates compared with using a higher granularity one [47,69]. Therefore, both time frame size and unit effects are error factors.

A comprehensive set of factors affecting estimates relates to the estimation process's particularities, such as the *use of historical data*. A field study connected it with a lesser magnitude of effort overruns [24]. A relevant number of studies also reported that the lack or no use of historical data is related to errors and problems in estimating — with evidence coming from respondent studies [99,159], laboratory studies [139], and field studies [21,57,100,110,133].

The *combination strategy of individual estimates* rose as a factor in our SLM, either for combining single values or interval estimates — with minimum and maximum values. We found evidence for three strategies regarding single values: statistical combination, unstructured group estimates, or Planning Poker. Three articles report evidence in favor of estimating in groups over averaging: unstructured group estimates [126] and Planning Poker (a structured approach) [25,125] led to less optimistic estimates compared with the average of individual estimates. When combining interval estimates, the results also favor group discussion over averaging [77]. Mahnic and Hovelja [111] found the same result for Planning Poker estimates compared with the statistical combination, but only when the participants in their experiments were software professionals. They found the opposite effect when students were estimating. In another study, the results suggested that planning poker is more accurate when the team has previous experience from similar tasks compared to unstructured group

[8] The use of the word size here is for simplicity. A task or project is larger in the sense that it requires more effort to be executed/implemented compared to others.

estimation sessions [40]. In summary, there is evidence in favor of estimating in groups over averaging estimates in general and in favor of Planning Poker more specifically.

Padding also impacts estimates' accuracy. The inclusion of a large buffer to deal with unexpected events and/or changes in the specification is a reason for accurate estimates [57]. The greater the preference for projects completed within the estimates, the greater the padding frequency [97]. More evidence about it comes from the fact that the removal of padding by management is related to estimating problems [100,110] and is a reason for inaccuracies [99]. Nevertheless, it is reported as an intentional increase in estimates aimed at the holding back reserves, which gives it a negative denotation [26].

The *anticipation of project' participants skills* emerged as a relevant factor for estimators. The inability to anticipate the team members' skills, abilities, or characteristics is a problem for estimating [100] and a reason for inaccuracies [99,159]. The knowledge about who will execute testing allows for the definition of testing effort [132]. However, one study suggests that the team's knowledge of who will work on the project may increase underestimation bias [152]. It might be the case that anticipating the project participants' skills may not work for all contexts.

Another essential aid is the *use of checklists*, leading to a lesser magnitude of effort overruns [24]. A field study indicates that using a personalized checklist during the estimation process reduces the underestimation bias [154]. Such evidence indicates that the use of checklists is an accuracy factor. Also, the lack of checklists is a reason for estimation error [57], meaning that its absence is an error factor.

The lack of *involvement of technical staff* during estimating is a reason for inaccuracies in three respondent studies [3,99,159]. Other three studies [91,96,153] also reported that an *informal basis for estimating* is an error factor. Lederer and Prasad [96] considered informal bases for estimating, comparing similar, past projects based on personal memory, guessing, and intuition as reasons for inaccuracies. The other two studies emphasized the lack of formality of the estimation process as a reason for inaccurate estimates [91,153].

Four factors associated with estimators regard their experience and skills. The first one is the *estimation experience*. It is an effort predictor in the context of mobile development [1], and its absence hinders the creation of a meaningful estimate [133]. The second is *experience with similar/previous projects/tasks,* which is also an effort predictor [151] and a reason for accurate estimates [57]. The third factor is the lack of *estimation skills*, an estimation inhibitor [110] that can cause estimation problems [91]. The fourth is the lack of *training in estimation*, which hinders creating a meaningful estimate [133] and is a reason for inaccurate estimates [159].

The final factor related to estimators at the planning phase is *enough effort and resources spent on estimation*, which is an accuracy factor and, when lacking, an error factor.

On the one hand, a respondent study reports that spending enough time on estimating is a reason for accurate estimates [57]. On the other hand, making quick, rough estimates is not motivating and hinders creating a meaningful estimate [133]. Also, insufficient time, effort, or resources for estimating is a reason for inaccurate estimates [57,73,110,159].

Two factors intersect **all the phases**. One of them is the *overall experience* of the estimator. In one study, experts' experience (including total experience, company experience, project experience, and the number of projects expert has participated) predicted estimation performance, leading to less estimation error [90]. Therefore, the presence of overall experience improved accuracy. Additionally, other studies indicate that the lack of overall experience is an error factor, leading to unintentional distortions of software estimates in varying directions - reducing or increasing them [109], hindering software estimation and accuracy [113], being a reason for estimation error [57].

The other factor affecting all phases is *standards in estimation*. All evidence about it is related to its shortage, and all results point to it as an error factor. It has many facets, in any case. For instance, in one case study, participants revealed that the lack of methodology or guidelines and the lack of setting and review of standards is a potential cause of estimating problems [100]. A follow-up survey confirms that these are reasons for inaccuracies [99]. Also, no development of estimation standards and no record-keeping of estimates and actual results make it difficult to capitalize on lessons learned [110], and no documented estimation procedure hinders the creations of a meaningful estimate [133]. Researchers also report that the lack of appropriate software cost estimation methods and processes [159] and the lack of clear guidance for estimating [3] are reasons for the inaccuracy of estimates.

5.3. Management roles

We present the factors regarding management roles — including higher management, project managers, and the Software Engineering Process Group (SEPG) — in Figure 10. We explored some of them thoroughly in previous sections: longer projects (Section 5.1), enough effort and resources spent on estimation (Section 5.2), and standards in estimation (Section 5.2). We explore all the others in the current section.

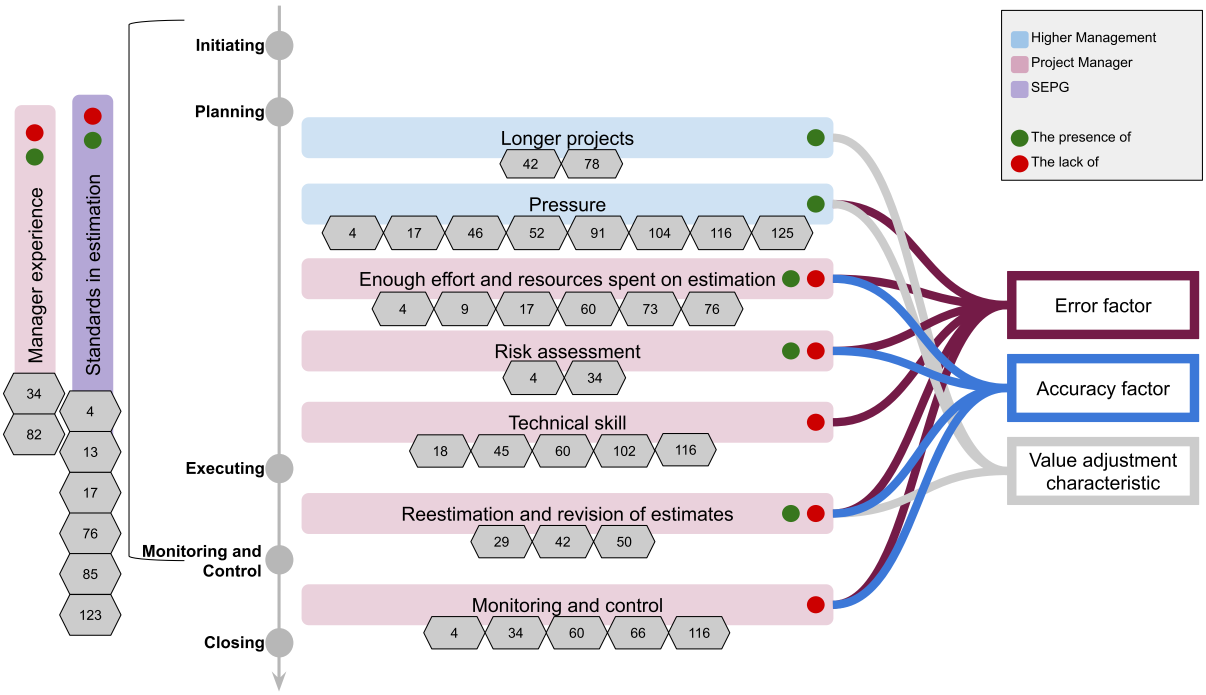


Figure 10 - Factors related to Managers

At the **planning phase**, *pressure* came up as an error factor. Yang et al. [159] report that the company's survival pressure and the business pattern are reasons for inaccurate estimates. Another facet of pressure is work pressure, which Altaleb, Altherwi, and Gravell report as an effort predictor [1]. Yang et al. [159] also inform that the senior manager's pressure to set or change the estimation results is a reason for inaccurate estimates - a finding that echoes in other studies [99,161]. It leads people to change their estimates intentionally [109], to cave in to people with more power [26], resulting in lower estimates than would be realistically expected [91]. A final facet is schedule pressure, which leads to more effort in test tasks [140] - and thus is a value adjusting characteristic.

Risk assessment is another factor in the planning phase. Systematic risk assessment related to lower error in duration estimates [127], and the lack of it is a reason for inaccurate estimates [159]. Surprisingly, some laboratory experiments' results indicate that identifying more risks immediately before software estimation leads to increased over-confidence [62]. Nevertheless, the authors stress that they have not investigated a complete risk management process - only the impact of simple risk identification.

Low *technical skills* also are among factors related to managers. One study report that Project managers not skilled in planning multi-disciplinary projects are reasons for estimation error [57]. Other studies also report technical skill issues but concerning the team, and we describe them further in Section 5.5.

At the **executing phase**, the only factor is the *reestimation and revision of estimates*. In a large company with two estimation points in its process, the reestimation at the analysis stage improves the accuracy of the effort estimates [152]. In a data study, more budget revisions were related to higher costs [94] - and therefore, we considered it a value adjusting characteristic. Nevertheless, in another data study, more estimation updates were connected with larger errors in effort estimates [95]. Regarding the last result, the authors explain that more extensive features had more frequent estimation updates. Another possible explanation is that projects already in trouble may undergo more estimation updates.

The only factor at the **monitoring and control** phase is its homonym and is an accuracy factor. One field study reports that good cost control is a reason for accurate estimates [57]. One a respondent study reports that adequate project administration is a reason for the prevention of overrun [32].

The factor that intersects **all phases** is the *manager's experience.* For instance, the number of projects previously managed correlates with duration error — more projects managed leads to lower error [127]. It is, therefore, an accuracy factor. Also, when the estimates used for the project contract are based on the project manager's previous experience only, it requires the developers to work over their capacity, which is a reason for low accuracy [2].

5.4. Technical roles

We found factors related to technical roles: requirement engineers, software designers, developers, and testers. Figure 11 brings such factors to the surface. None of them apply to all phases. We explained two of them in Section 5.1: changes to requirements or scope and pressure — including the factor associated with the **tester** role.

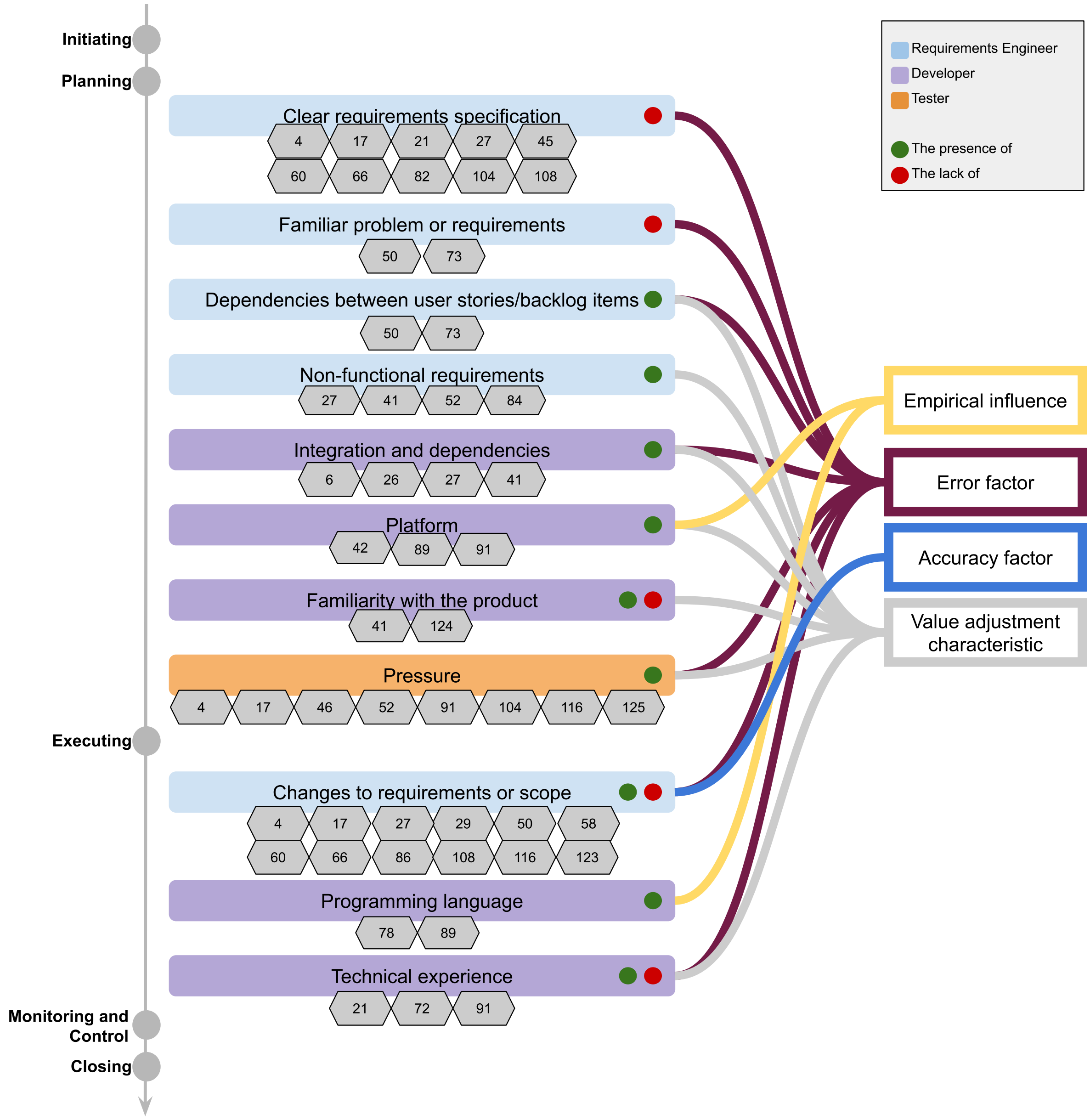


Figure 11 - Factors related to people in technical roles

We found four factors related to requirements at the **planning phase**, which we associated with the **requirements engineer** role. One of them is a clear *requirements specification*. Some studies present results in more general terms, indicating that poor, unclear, or ill-defined requirements are one reason for inaccuracies [32,57,99,153,159,161]. Other studies emphasize specific facets that make requirements poor, like the redundancy of user stories [21], missing requirements [153], weak or ambiguous requirements [24], incomplete requirements [57], and the user's lack of understanding of their requirements [7]. All this evidence indicates that the lack of clear requirements specifications is an error factor.

Familiar problems or requirements was also classified as an error factor when they are absent. Layman et al. [95] report that unfamiliar feature requirements are a reason for estimation inaccuracy. Jørgensen and Gruschke [73] report that too little knowledge about the problem is a reason for estimation inaccuracy.

The third factor associated with the requirements engineer is *dependencies between user stories/backlog items*. Conoscenti et al. [21] found that links to other stories serve as indicators for a possible inaccurate estimation. Altaleb, Altherwi, and Gravell [1] found that dependency between backlog items is an effort predictor in the mobile development context.

The fourth factor we found regards studies reporting that *non-functional requirements* are an effort predictor or a cost driver [151,153]. We also found studies reporting that specific non-functional requirement types are associated with higher effort, like the high legal or regulatory impact of the code [101], the required level of performance, and the required security level [140]. So, we classified it as a value adjustment characteristic.

Still in the **planning phase**, three factors emerge for the **developer** role. One of them is *integration and dependencies*. One study report that technical dependencies are an effort predictor in agile global development [17]. Another one considers that integration issues are a cost driver, also in the context of agile development [153]. In the context of corrective maintenance of object-oriented systems, a high level of code/system dependencies leads to higher effort [101]. Therefore, the integration and dependencies factor is a value adjustment characteristic. Another study informs that integration complexity is an estimation challenge [108], suggesting it is also an error factor.

The other factor regarding developers is the *platform*. In the context of mobile development, the supported Platform type (IOS/Android./Win./etc.) and the supported device (phone, tablet, smartwatch) are both effort predictors [1]. Other two studies report that the type of platform impacts software costs [94] and that the interaction of team size and development platform has a significant impact on productivity [45].

Finally, the developer's *familiarity with the product* is a value adjustment characteristic. When low, it leads to more need for effort [101]. In another study, the programmer's familiarity in the number of months of experience with the system was a significant predictor of debugging effort (more experience leads to less effort) [22].

Two data studies inform the *programming language*'s importance as an empirical influence over the estimates related to the **developer** role at the **executing phase**. It has a significant impact on the effort needed [41] and on time-to-market [45]. Huang, Sun, and Li [45] also report that team size and language type interaction significantly impact productivity.

The technical experience related to the **developer** role is an additional factor we found. Altaleb, Alterwhi, and Gravell [1] evidence that developer implementation experience is an effort predictor. Also, developers' lack of experience leads to estimation inaccuracy [21], and the lack of technology experience leads to a higher probability of effort overrun [39].

5.5. Team

Some of the factors we found regarded the whole software team. We show them in Figure 12. We thoroughly discussed two of these factors in previous sections: *involvement of technical staff* in estimating and *experience with similar/previous projects/tasks* – both at Section 5.2.

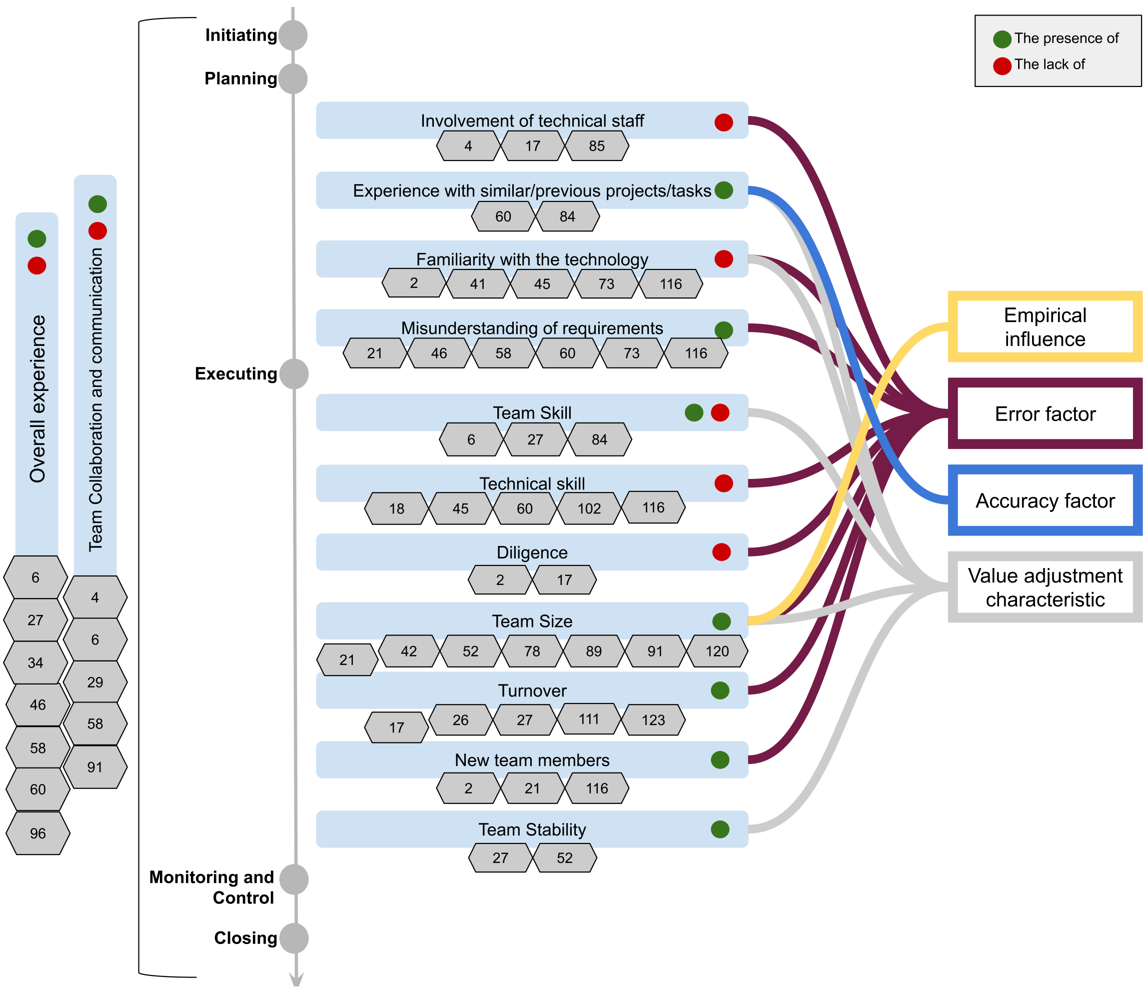


Figure 12 - Factors related to the team

At the **planning phase**, *familiarity with the technology* is a value adjustment characteristic because when it is low, it leads to a higher need for effort [101]. Other studies also indicate that the use of new or little-known technology is a reason for estimation inaccuracies [9,24,73] and a significant threat to estimates [91]. Also, many studies report results regarding how the *misunderstanding of requirements* leads to estimation inaccuracy and errors [21,57,73,91,113]. It also causes unintentional distortions of software estimates in different directions: either as increases or decreases of estimates [109].

The team's *skill* is a value adjustment characteristic at the **executing phase** once three studies present it as either an effort predictor or a cost driver [17,151,153]. Another more specific factor is the *technical skill*, which we partially addressed in Section 5.3. The presence of unskilled members in the team is a reason for inaccurate estimates [153]. Lack of technical skills [24] and technical expertise in a particular area [91] lead to estimation inaccuracies. Less software development skill is weakly connected with optimistic predictions too [71]. More specifically, Jørgensen, Bergersen, and Liestol [53] reported that lower programming skills connect with higher over-optimism in larger tasks, higher over-pessimism in smaller tasks, and higher over-estimation in smaller tasks.

Two respondent studies report how *diligence* issues may impact estimates negatively. Lack of diligence by systems analysts and programmers is a reason for inaccuracy [99]. Also, the delay of decisions concerning requirements due to team members' lack of responsibility and motivation is a reason for a higher need for effort than estimated [9]. So, lack of diligence is an error factor.

Many studies report findings regarding a range of issues related to team's size and stability issues. The team's *size* is an effort predictor [1,45], and larger teams connect with higher effort and costs [41,94,140]. It is, therefore, a value adjustment characteristic. One of these studies also suggests that the interaction of team size and language type and the interaction of team size and development platform significantly impact productivity [1,45]. Interestingly, two studies suggest that multiple developers' involvement in a story or a task may lead to over or underestimations [21,43]. So, larger team size also is an error factor.

The last three factors of the executing phase are intricately connected. *Turnover* is a reason for inaccuracies in estimates [99,103,153] and estimating problems [100]. The loss of organizational knowledge due to high turnover is an estimation challenge [108]. The existence of *new team members* leads to estimation inaccuracies [21] and a higher need for effort than estimated [9]. Another study reports that the introduction of new people is a major threat to accurate estimates [108] - and therefore, we classified it as an error factor. Finally, regarding team *stability*, one study reports it as a cost driver [153], while another one stresses that team continuity leads to less effort in the context of testing tasks [140]. Therefore, team stability is a value adjustment characteristic that estimators should account for when estimating.

Two factors impact **all phases**. The *team's overall experience* is one of them — and we explored some of its facets in Section 5.2. Three studies report it as more specifically connected with the team. Two respondent studies put the team's overall experience as an effort predictor or a cost driver [17,153]. Another respondent study indicates that low team experience correlates with duration error, with less experience leading to more error [127].

The other factor related to all phases is *collaboration and communication*. The communication process and the communication model are effort predictors [1,17]. On the one hand, team collaboration facilitates software estimation and accuracy [113]. On the other hand, the lack of stakeholder collaboration is a reason for inaccurate estimates [159]. Also, inherent difficulties related to communication and coordination present in multi-site arrangements lead to higher effort overruns [152].

5.6. No specific role

In Figure 13, we present a whole set of factors we found that is not specifically connected with any roles. They may impact or be caused by any or all of them.

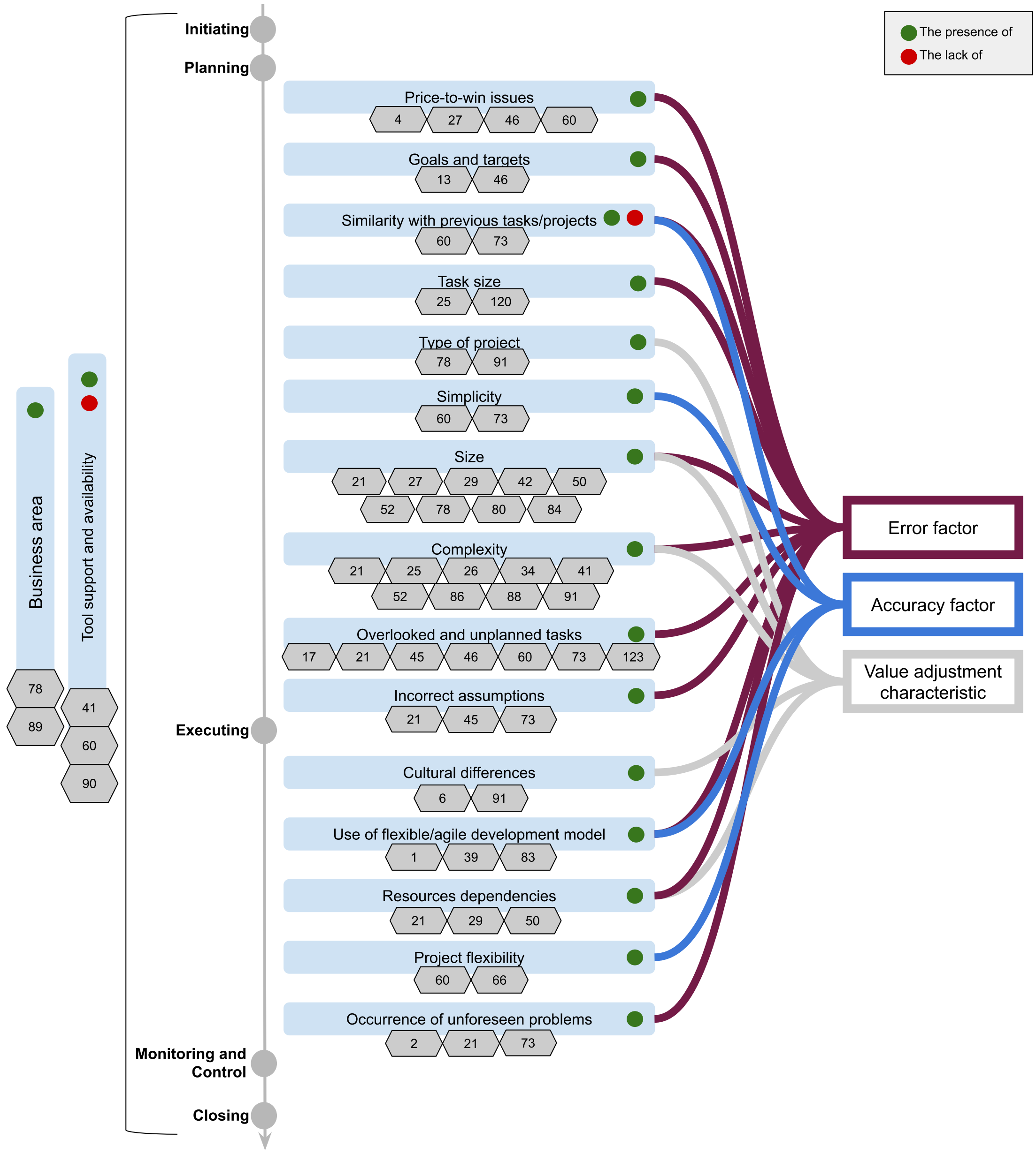


Figure 13 - Factors unrelated to any specific role

During the **planning phase**, *price-to-win issues* play a role in estimation when present. Price-to-win is described as an estimate defined by the price or schedule needed to win a job [14]. An estimate strongly impacted by price-to-win is a reason for estimation error [57]. Allowing the project bidding requirements to predefine the project cost [159] or purposefully underestimating the effort to obtain a contract [153] are reasons for inaccurate estimates. Magazinius, Börjesson, and Feldt [109] also report intentional distortions of software estimates in varying directions because estimates are budget determined. Somewhat related is the *goals and targets* factor. In field studies, the authors report that personal goals affect the estimates

[110], and that personal or organizational agendas lead to intentional distortions of software estimates in varying directions [109].

We identified that some of the project and task characteristics also are relevant factors for estimation, such as the *similarity with previous tasks/projects*. On the one hand, a task similar to the ones previously completed is a reason for improving estimation accuracy [73]. On the other hand, projects frequently different from earlier projects are a reason for estimation error [57]. The *task size* is also an error factor: larger tasks are more prone to effort overruns [154], and tasks with more subtasks were underestimated compared to tasks involving fewer ones [43]. Another characteristic that emerged as an effort predictor is the *project type*: whether it is related to a new or enhanced application in mobile development [1]. He et al. [41] also report that the enhancement projects may consume the most effort . Simultaneously, re-development may need less effort than enhancement, and new development may consume even less than re-development. Therefore, the project type is a value adjustment characteristic. Finally, two studies inform that task or project *simplicity* is a reason for accuracy [57,73].

A subset of the planning phase factors regards the product characteristics: the product *size* and *complexity*. *Size* is a value adjustment characteristic since many studies report it as a cost driver, effort predictor, or as correlated to effort [41,151,153,157] — with larger project sizes leading to more effort [94]. Size is also an error factor. For instance, Conoscenti et al. [21] report that user story size serves as an indicator for a possible inaccurate estimation. In a data study, more extensive features correlated to larger errors in effort estimates [95]. Finally, a field study indicates that smaller product customizations tend to be overestimated, while larger ones tend to be underestimated [152].

Complexity is a factor with many facets. Requirements complexity [140] and high technical complexity [101,140,143] leads to more effort. In the context of mobile development, one study points out that application form complexity is an effort predictor [1]. Therefore, complexity is a value adjustment characteristic. Some studies report technical complexity [108,127,154,160] and feature complexity [21,108] as estimation challenges or as related to inaccuracies, delays, and under or overestimations.

Overlooked and unplanned tasks is another impacting error factor: it is a challenge for estimation [100] and a source of inaccuracies and errors [21,24,57,73,99,109]. Unplanned tasks or re-work also is a reason for estimation error [57]. Closely relate4d, *incorrect assumptions* when estimating is also an error factor that may be related to the code [73], functionality [21], or complexity [24,73].

At the **execution phase**, distributed development issues also play a role when they are present. Two studies report *cultural differences* as an effort predictor [1,17]. Thus, estimators should consider it a value adjustment characteristic if there are multiple development sites with differing cultures.

The *use of flexible/agile development models* is an accuracy factor regarding project and task characteristics. Moløkken-Østvold and Jørgensen [124] report that flexible models are associated with lower effort overruns than sequential models. Koch and Turk [93] also report that the use of agile methods is related to less effort deviation from estimate than rigid models. However, Brown et al. [18] inform that software developers give lower estimates when the development method is agile than when the development method is a waterfall, suggesting their estimates were optimistic.

Resources dependencies also stood out as one factor affecting estimates. Depending on external resources may lead to delays and/or higher effort that should be considered when estimating [63]. Also, dependencies (such as for code reviews) on specific human resources (e.g., product architects) introduce delays [152], and developer resource constraints and external commitments are a reason for estimation inaccuracy [95].

Project flexibility is another relevant accuracy factor: a high degree of flexibility in implementing the requirement specification is a reason for accurate estimates [57]. Another study reports that project flexibility to reduce the scope (functionality/quality) in order to meet plan and budget is a factor more frequent in projects with lower overrun (less than 20% overrun) compared to projects with higher overrun (more than 20% overrun) [32].

The *occurrence of unforeseen problems* is a factor that impacts estimates negatively. The occurrence of risks, unexpected events, or technical problems leads to a higher need for effort than estimated and estimation errors [9,21,73].

Two of the factors affect **all phases**. The *business area* has an impact on the effort [41] and productivity [45]. The other factor is *tool support and availability*. Software development tools have an empirical influence over management and testing efforts [144]. Additionally, insufficient tool support for project management is a reason for estimation error [57], and the low availability of required tools leads to higher effort [101].

6. Discussion

Our primary research question for this SLM was ***RQ 1 - What is the existing evidence about the factors that affect expert judgment software estimates?*** In this section, we summarize our current answer to this question and discuss our findings.

6.1. The seas of factors that researchers explored the most

The top-five factors in the SEXTAMT regarding the number of articles reporting them are *changes to requirements or scope* (12 articles), *clear requirement specifications* (10 articles), *product size*, *complexity*, and *use of historical data* (9 articles each). Most factors

(40, representing around 58% of the total) were reported in three or more articles. The remaining 29 factors (around 42%) were investigated in two research articles only, indicating that they could benefit from further investigation.

In addition, many of the top factors were probably investigated extensively because of their true impact on the estimates. Nevertheless, others may have been investigated because of a controversial result. Controversies possibly exist either because of differences in research design or because such factors are more sensitive to the context. Future research efforts can aim to clarify which is the case. For instance, regarding the *combination strategy of individual estimates* most of the results shows that group estimation led to less optimistic estimates compared with averaging. However, one study found the opposite when participants were students [111]. It is unclear whether this controversial result is due to the difference in choice of participants (software professionals or students) or whether experience interacts with the combination strategy to define which one will bring superior results (more on this in Section 6.5).

Also, if a factor is shown to influence estimates through the employment of varied research strategies, we can more confidently believe that such an effect exists. Each research strategy has its inherent limitations and strengths [141]. Also, each one has the potential to maximize one research quality criteria at the expense of others. For instance, respondent strategies have the potential to maximize generalizability; field strategies can maximize realism; laboratory strategies can maximize control; and data strategies can maximize precision [142]. Therefore, we evaluated the existing evidence for the factors in the SEXTAMT by considering the research strategies that researchers employed to investigate them.

Figure 14 represents only the factors investigated in five or more articles — 21 factors in total, represented by the light gray edges surrounding the top of the circle. We also mapped the factors to the research strategies that researchers employed to investigate them, represented at the bottom of the circle: respondent (R, in dark red), field (F, in blue), data (D, in dark gray), or laboratory (L, in orange). Next, we discuss the type of evidence derived from such studies, considering all these research strategies.

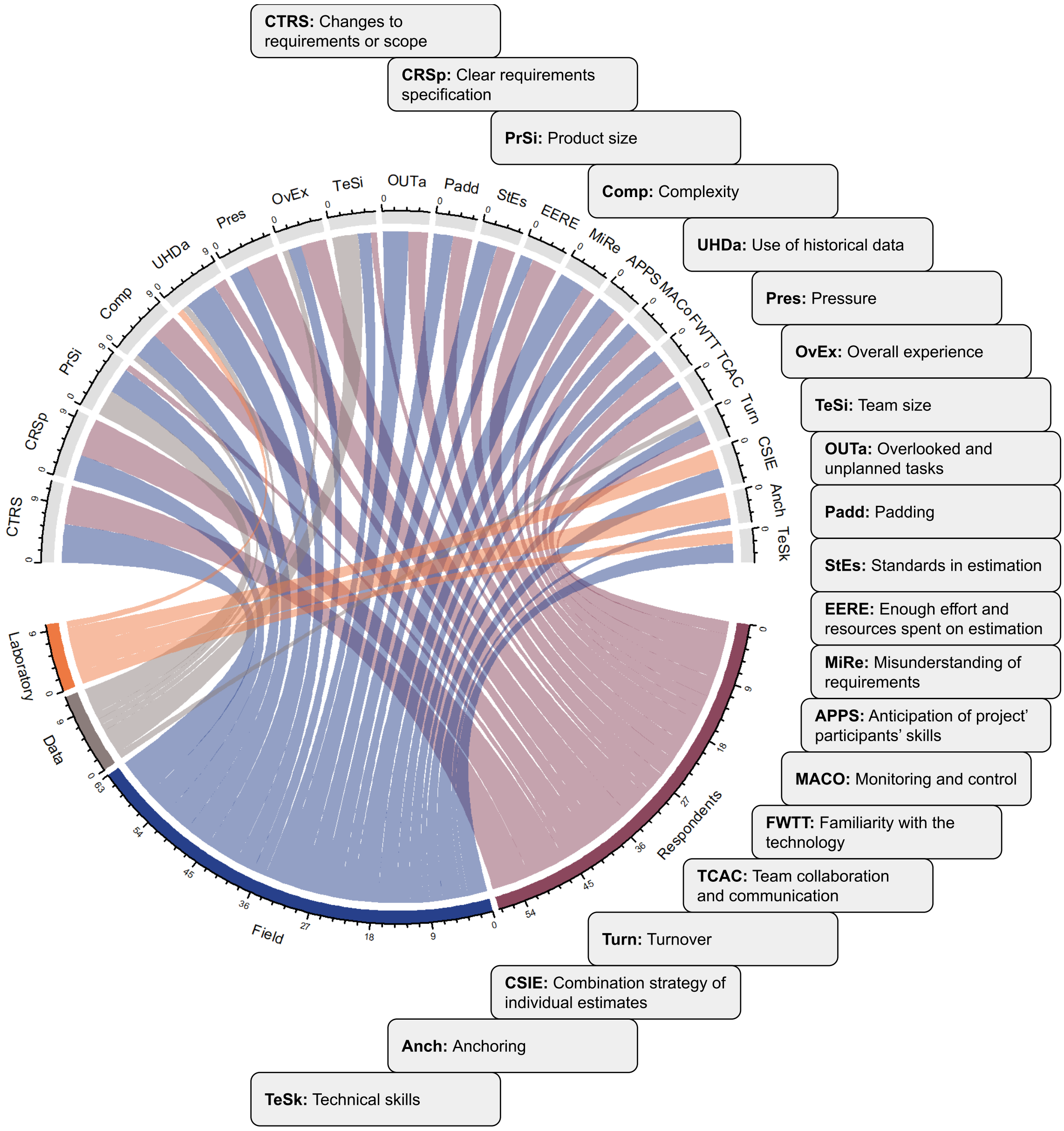


Figure 14 - Top factors and the studies' research strategies

First, six factors have been investigated employing at least three different research strategies: *product size* (1 R, 4 F, 4 D), *complexity* (4 R, 4 F, 1 D), *use of historical data* (2 R, 5 F, 1 D, 1 L), *overall experience* (4 R, 2 F, 1 D), *team size* (1 R, 2 F, 4 D) and *turnover* (2 R, 2 F, 1 D). Most of them were investigated through a combination of research, field, and data strategies — suggesting the generalizability, realism of context, and precision of data regarding the supporting findings. Some of these factors are classic cost drivers, such as *product size* and *complexity*, and software companies may not have much control over them. Other factors are more controllable but may not be so easy to implement. Still, software practitioners and organizations can organize themselves to *use historical data* when estimating, increase their *overall experience*, regulate *team sizes* to keep them small, and reduce *turnover*.

All the remaining factors in Figure 14 were investigated using two different research strategies. In summary, these factors indicate that improving the estimation process is necessary, but not enough to get better results. Practitioners need to enhance the requirements engineering and management process. For example, we need to work on getting *clear requirements specifications* (6 R, 4 F). Moreover, human and social aspects play a crucial role, as highlighted by factors pertaining to the categories of political issues (such as *pressure* — 5 R, 3 F), experience (*familiarity with the technology* — 3 R, 2 F), skill issues (*technical skill* —3 F, 2 L), biases (*anchoring* — 1 F, 4 L), and team issues (*team communication and collaboration* — 4 R, 1 F). Unexpected events also have their role: *overlooked and unplanned tasks* (3 R, 4 F). Reducing such events is necessary — possibly with the *use of checklists*, another factor from SEXTAMT.

The SEXTAMT factors excluded from Figure 14 were reported in four or fewer articles and investigated through no more than two research strategies. They can further enrich our understanding of the impact of the requirements and the estimation process, for instance. Nevertheless, they expand our perspectives to other directions as well, such as the impact of product characteristics, client and user issues, environment, attitudes and maturity, and testing and rework.

In any case, software organizations and practitioners aiming to diagnose the factors more relevant to their context to improve their estimation results can use the SEXTAMT factors to guide what to include in internal surveys, for instance. Practitioners can also use the SEXTAMT factors (especially those classified as value adjusting characteristics) to build internal checklists. For instance, Usman et al. [154] proposed a process to build checklists to support expert judgment estimation, and the first step is to understand the estimation context. This step has the objective to elicit the factors that should be included in the checklists by, for instance, surveying the literature on the search for effort or cost drivers. The SEXTAMT already provide a map of such factors, and practitioners can save time by using it instead of surveying the literature themselves — a process that involves high costs.

In addition, some of the SEXTAMT factors can be helpful in the debiasing strategy that Kahneman, Sibony, and Sunstein [87] proposed to help improve judgments in general: decision observers, i.e., people in charge of observing others making judgments in real-time to identify and alert on the occurrence of biases. Decision observers use checklists to accomplish their tasks, which should be adapted to their specific domain. The SEXTAMT factors can guide such adaptation to the software estimation domain. Particularly, the factors from the bias and the estimation process seas at SEXTAMT can provide valuable items.

Also, practitioners can use the SEXTAMT factors as input for risk analysis for their projects, improving their project planning, monitoring, and control. For instance, projects planned to deliver more extensive or more complex products, with less experienced software teams, or where estimators cannot anticipate the participants’ skills when estimating run a larger risk of estimating error and, therefore, of failing to meet their commitments. Thus, project managers of such projects need to be especially caring for monitoring these factors.

> **Takeaway message 1:** There is solid evidence for the factors in the SEXTAMT, with 40 of them reported in three or more articles. A few of those — six in total — were investigated by applying at least three different research strategies. The remaining 29 factors were reported in two studies each, suggesting they can benefit from further investigation.
>
> **Takeaway message 2:** Practitioners can use the SEXTAMT factors (i) to help diagnose the factors more relevant to change in their contexts, in software process improvement initiatives; (ii) to build supporting checklists for their estimation activities when using expert judgment; (iii) to improve their estimation results in real-time as part of debiasing interventions; or (iv) as input to risk analysis of software projects.
>
> **Takeaway message 3:** Practitioners interested in improving their estimation can rely on the existing evidence that points to the need for improving the requirements engineering and the estimation process, but also indicates the necessity of considering factors associated with political issues, the management process, experience, team issues, biases, and technical skills.

6.2. Looking through the lenses of the temporal and stakeholder dimensions

When it comes to the process phases in which factors cluster, the planning and executing phases are the ones that stand out. It is natural to have factors at the **planning phase**, because estimating occurs primarily during such stage. At the **executing phase**, factors emerge because the dynamics of projects impact estimating error and accuracy. For instance, our software projects have a moving target [109], and we found in our SLM that *changes to requirements or scope* are an error factor, especially if the original estimates are not modified to reflect the changes. *Overlooked and unplanned tasks* may also be revealed by project execution dynamics, leading to a higher need for effort, costs, and duration than expected.

It is noticeable that only one factor emerged at the **initiating phase** and none at the **closing phase**. However, when looking for the factors reported in one article only, we can find more about such phases. For instance, *bidding situations* are relevant at the **initiating phase**, with one field experiment reporting that companies selected on the criteria of the low bid have higher cost overruns, a phenomenon known as the “winner’s curse” [54]. Therefore, estimators might need to pay special attention to the initiating phase in bidding contexts.

Additionally, more investigation on learning and feedback has the potential to shed some light on what is relevant at the **closing phase**. For instance, at least four studies ([57], [113], [71], and [73]) suggest that estimation error, feedback, and learning from past projects and tasks might be beneficial to reducing overconfidence and improving estimates.

Regarding stakeholders, many of the factors are related to **estimators**, which is expected once they are the primarily responsible people for estimates. Our results also indicate the power of other roles that might not be directly involved with the estimating process, such as the client and managers.

> **Takeaway message 4:** Most factors cluster at the planning phase, because estimating occurs primarily at this stage. Many factors also pertain to the execution phase because project dynamics can alter the assumptions on which estimates were generated.
>
> **Takeaway message 5:** The initiating and closing phases are less explored, and we can benefit from investigating more factors regarding such phases.
>
> **Takeaway message 6:** Many factors are related to estimators, and many others indicate the power that people playing other roles also have over the estimates, showing that improvement initiatives in the industry must account for them too.

6.3. The strategies researchers employed to explore the seas

As for the project variables, most studies focused on **effort**, which is understandable - as McConnell [116] suggested by his flow of well-estimated projects that the effort is an intermediary estimate in software projects, ideally used as input to cost and duration estimates. Therefore, factors that impact effort estimates indirectly impact both cost and duration, and because of that, researchers may consider it more beneficial to focus on them.

The mechanism for measuring the impact of the factors that researchers applied the most is rather indirect: the **participants' perceptions** of reasons for errors and accuracy. Such an approach may provide rich insights into the phenomena that cause errors when estimating or promote accuracy in field settings. Considering that many participants in respondents and field studies in our SLM are experts in software development and maintenance tasks, we cannot overlook their opinions about the factors affecting estimates. Nevertheless, the approach has drawbacks also. For instance, people may attribute different meanings to the term "estimate", even when they work at the same company [67], making it difficult to interpret the results of surveys [59].

Another widely employed mechanism for measuring the impact of factors over the estimates was the **difference of estimates** between groups. The difference of estimates does not provide direct evidence about accuracy, but it can evidence when a factor causes an estimate to increase or decrease for reasons beyond the estimation process. This allows us to identify factors that can induce optimism in estimators, leading them to provide low estimates instead of realistic ones. Considering that extensive projects tend to be underestimated with a median time overrun of 20% [38], identifying such factors can be very useful.

Additionally, researchers have used objective error measures, such as **MRE, MREBias, BRE,** and **BREBias**. Nevertheless, since the 90's at least, MRE has been criticized because it has the disadvantage of weighing differently under and overestimations. Underestimations are not weighted sufficiently, leading to higher penalization of overestimations [59]. MREBias suffer from this same problem. BRE and BREBias are balanced metrics in this sense [124]. In Figure 15, we grouped MRE and MREBias under the label "Unbalanced" and BRE and BREBias under "Balanced". It shows that, gradually, researchers are moving to the use of more balanced metrics over the years.

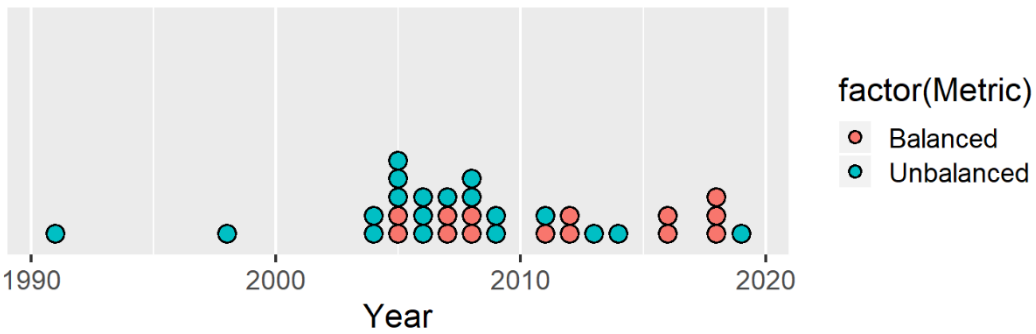


Figure 15 - Balanced (BRE & BREBias) x unbalanced (MRE & MREBias) over the Years.

Also, researchers prefer accuracy metrics over bias: with 19 occurrences for MRE and BRE together versus 15 occurrences of MREBias and BREBias. Accuracy is the average unsigned error, irrespective of whether the estimate is too high or too low; bias is the average tendency to generate too high or too low estimates [37].

In any case, using MRE or BRE and similar metrics can be misleading because they depend on actual values, and work can be adjusted to fit an initial estimate [80], leading to a "moving target problem" [59] and to a distorted perception of accuracy. For instance, this makes it harder to understand exactly whether a factor contributed effectively to improving estimation accuracy, or whether a software team just took advantage of a higher project flexibility to create an illusion of accuracy. A possible solution comes from the literature about judgment in general: the measurement of noise instead of bias or accuracy. Noise is the random scatter of judgments that should ideally be identical — or in other words, unwanted variability, a significant component contributing to judgment error, along with bias [88]. The advantage of measuring noise over bias or accuracy is that we do not need to know actual values. One issue that emerges from this discussion is how to measure noise. A common measure from statistics is the standard deviation [89].

Nevertheless, we found very few studies discussing the variability of estimates in the software domain. Only one study explores explicitly the issue, showing a high level of inconsistency when software practitioners estimate the same task, based on the same information and under the same conditions, but at different times [33]. In addition, very few studies in our SLM report the standard deviation of estimates, when using the difference of estimates as a measurement strategy (see [138] and [130]). This reveals a low awareness of researchers in our community regarding noise, its relationship with error in expert judgment estimation, and the benefits of measuring and reducing it. Regarding software estimation practice, it is unclear whether practitioners share the perspective of researchers about this concept. In any case, software organizations can benefit from investigating how much disagreement there is among their professionals estimating the same tasks independently.

Regarding research strategies, researchers employed the **laboratory research strategy** widely, and the **respondents' strategy** was quite popular too. Laboratory research strategies favor the investigation of only a few factors at once. In contrast, the articles employing respondents strategies tended to reveal much more factors in each study, contributing significantly to the wide variety of factors we found. The factors with more articles using a laboratory experiment strategy were also the ones that researchers refined the most by investigating relevant variations. For instance, researchers investigated different nuances of the *anchoring* effect, assessing the impact of both numerical and textual anchors [56], as well as of single and interval anchors [105]. Another refinement was the investigation of the moderating effect of the expertise of the source and of the receiver of the anchor value [105] and the impact of one intervention to reduce its effects [138]. Another example is the *sequence effect*, whose impact over the estimates varies with the size of the tasks estimates in the sequence [75]. Researchers perceived an assimilation effect (the estimate become more similar to the one of a previously estimated task) for tasks of different sizes, and a contrast effect (the estimate become more different than the previous one) for tasks of similar sizes.

When considering the taxonomy of Stol and Fitzgerald [141] for research strategies, it is interesting to notice that the studies employing the field strategy, there are very few field experiments - a total of 10. In other words, when it comes to factors affecting estimates, researchers are more likely to enter natural settings to collect data without manipulating variables. Probably such manipulations are hard to be approved by administrative staff or to be adequately carried out. Thus, they restrict the manipulations of variables to the lab, reinforcing the need for triangulation of strategies [141] to evaluate further the impact of factors investigated.

Additionally, considering that the potential for generalizability from respondent studies and the potential for realism from field studies can be taken as proxies of the relevance of research results for practice, from all the 69 factors from the SEXTAMT, most (62) have this type of evidence. From the seven factors with no evidence from respondent or field studies, three are related to biases on estimation and were investigated through lab studies only: *sequence effects*, *time frame size*, and *unit effects*. The *client's expectation* was a factor investigated only through lab studies. The *programming language*, *business area*, and *longer projects* emerged from data studies only. Nevertheless, the lack of evidence from respondents and field studies for these factors does not mean they are irrelevant. For instance, practitioners are not aware of the biases affecting them in many cases, which makes it impossible for them to point this kind of factor in respondent studies. Therefore, combining research strategies reveals complementary findings in research topics so complex as this one. This has been highlighted before in the study of reasons for software effort estimation error in one single company: combining information sources, data collection methods, and data analysis methods leads to complementary insights [57].

Takeaway message 7: The participants' perceptions can provide a rich picture of factors affecting estimates in practice, even though it provides a subjective perspective. For more objective measurements of impact, the difference of estimates between a control and an experimental group has been largely adopted.

Takeaway message 8: Despite the criticism over metrics such as MRE, researchers are still gradually moving to use more balanced metrics such as BRE to assess the accuracy of estimates.

Takeaway message 9: Researchers are not fully aware of the concept of noise and its contribution to estimation error, even though it can reveal estimation problems with the benefit that we do not need to know actual values to measure it. It is not clear whether practitioners are unaware of it as well. In any case, software organizations can benefit from noise audits as starting points to improvement initiatives and noise measurements to assess the effectiveness of interventions to their estimation processes.

Takeaway message 10: Respondents strategies allowed for discovering many factors relevant in practice, while laboratory strategies allowed for the refinement of factors.

Takeaway message 11: The combination of different research strategies provides complementary factors, allowing for a richer map of the factors affecting expert judgment estimates.

6.4. Into the wild – part 1: underexplored seas

We excluded from the SEXTAMT a total of 166 factors reported in one research article only each[9]. Therefore, we consider they are in a gray area, and there is a need to execute more research to strengthen the evidence about their impact. Some of them have the potential to enlarge the territory of existing seas in the SEXTAMT. In contrast, others have the potential to reveal new seas of their own.

Such a myriad of factors shows that researchers have investigated a set of varied factors affecting expert judgment software estimates. However, this does not mean that all factors reported by unique articles are worthy of further investigation. We need some filtering on them to decide which ones are good candidates for more studies. For instance, *luck* is a factor reported in a respondents study. However, what does it mean? Also, the presence of other factors we identified in our SLM might explain luck to some extent: we can consider that a software project was luckier because requirements did not change, for example.

In addition, we classified some of these factors as a satellite to others, meaning they are somewhat related, even though not enough to be united to create a larger one. One example is “team process experience” and “expertise of new team members”. Although they are related, the first can relate to all team members, not only to new ones, while the latter is very specific in including only new people. Therefore, we cannot unify them to form a single factor investigated in three articles, allowing its inclusion among the SEXTAMT factors. Therefore, we kept them as part of the unique factors, marking them as satellites of each other.

Another example is the case of the factor *forcing to stay within the estimate*. It is a satellite of one SEXTAMT factor: *project flexibility*. For instance, software practitioners need the flexibility to deliver less polished features when they are forced to stay within a deadline, no matter what. We can also argue that *forcing to stay within the estimate* is a repercussion of other factors from unique articles, such as *estimates interpreted as commitments* or the *use of uncertain estimates as baselines*. Nevertheless, researchers have not validated such relationships. In any case, we indicate the satellite factors as part of our supplementary material.

Factors investigated through laboratory research strategies are good candidates for field experiments to assess whether their impact is kept in real-life contexts. Take the *format* factor, for example, which is about using the traditional request format — “How much effort is required to complete X?” – versus using an alternative format — “How much can be completed in Y work-hours?”. In the laboratory, the alternative format has led to more optimistic estimates [74]. However, it is precisely the format we expect when using agile methodologies. Does it impact estimates negatively in the trenches, making them more

[9] A complete list of such factors, together with their codes and categories can be found here.

optimistic? Another factor whose effect is relevant in the same context is the *use of Fibonacci scales* that, compared to linear scales, led to lower estimates when using Planning Poker [147].

> **Takeaway message 12:** Researchers have investigated a large and varied set of factors affecting estimates when using expert judgment. Most of such factors were reported in one article only, needing more research to strengthen the evidence about their impact.

6.5. Into the wild – part 2: validated relationships among the factors

The discussion of satellite factors leads us to another underexplored issue: the relationship among different factors. Therefore, after answering the main research questions that we presented in Section 3.1, we decided to extract and analyze data for an additional question: "*SQ 1.6 – What are the validated relationships among the factors affecting expert judgment expert estimates?*"

Only nine articles had results regarding such relationships. We illustrate the relationships we found in Figure 16, where each light blue rounded rectangle represents one SEXTAMT factor. Each gray rounded rectangle represents one factor we did not include in the SEXTAMT because it was investigated in only one article.

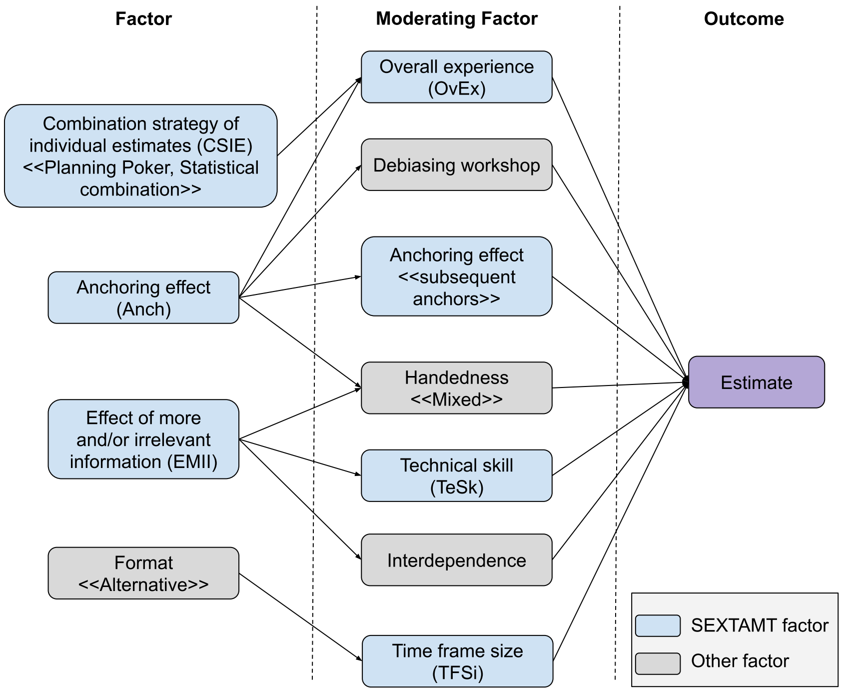


Figure 16 - Relationships among factors.

Figure 16 shows that *overall experience* moderates the impact of the *combination strategy of individual estimates*: more experience is connected with less optimistic estimates when using Planning Poker compared to when using a statistical combination of estimates. In the context where estimators have less experience, the result is inverted: the statistical combination leads to less optimism [111]. This result suggests that without experience discussions lack the benefit of meaningful divergent perspectives about the task complexity, or the wisdom to recognize forgotten tasks, or other flaws in judgment. It also seems that higher experience is needed to overcome the effect of the social influence bias [106] in group discussions of the estimates. Nevertheless, we should consider this result carefully because researchers contrasted a sample of students (representing less experience) with a sample of software professionals (representing more experience).

Overall experience also reduces the impact of the *anchoring effect* over the estimates [105], as well as *debiasing workshops* [138] and the use of subsequent anchors aimed at neutralizing first impressions caused by the first anchor [76]. However, in none of these studies, the anchoring effect was completely removed: only reduced. In another study, the researchers showed that mixed-handers were more strongly influenced by anchors compared with strong-handers [58], revealing how *handedness* can influence estimates.

Handedness also impacts the *effect of more and/or irrelevant information*: mixed-handers also are more impacted by irrelevant information [58]. In addition, people who score high in *interdependence* are also more strongly influenced by more and/or irrelevant information than people who score low [56]. Higher interdependence refers to higher connectedness to others and higher importance to social context and relationships. In addition, another study showed that higher technical skill reduces the impact of more and/or irrelevant information [72].

Also, the *time frame size* moderates the impact caused by using an alternative *format* for requesting estimates. Smaller time frames increase the impact by leading to more optimistic estimates [35,74].

Interestingly, many of the factors on the leftmost side of Figure 16 are related to a psychological or social bias. For instance, combination strategies of individual estimates are subject to social biases. The anchoring effect is a psychological bias. The presence of more and/or irrelevant information can also bias judgment, leading people to think the task is larger than it truly is, for example. Additionally, the moderating factors give us hints about interventions to deal with such biases. For instance, if we know estimators have low experience, it might be wiser to use the statistical combination of estimates instead of Planning Poker. Another example is composing estimating teams to include people with higher experience and technical skills whenever possible because this helps reduce the effects of psychological biases.

These results show the relevance of studying the relationships among factors. Such relationships can reveal the paths of interaction among factors and the ones that can trigger chains of negative or positive effects over the estimates. Therefore, in software process improvement initiatives regarding the estimation process, focusing in factors mediating or moderating others can be a cost-effective strategy to improve accuracy.

> **Takeaway message 13:** Although the set of factors affecting estimates when using expert judgment is large and varied, we could benefit from more studies exploring the relationships among such factors. This investigation can help narrow down the factors to focus to keep a good balance of costs and benefits when dealing with estimation problems.

7. Threats to validity

We analyzed the validity threats to this SLM, considering threats to the study selection validity, threats to data validity, and threats to research validity [4]. 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. Another mitigation action to this threat was the use of a known set of papers to evaluate the search strategy [162]. The goal of this evaluation was to reach a sensitivity of 70% in automated search [162]. A final mitigation action to this threat was snowballing procedures to enlarge the number of retrieved relevant papers, reaching a sensitivity of 100% afterward. 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. Additionally, the authors executed the selection process over a sample of the articles, discussing any inclusion or exclusion conflicts. Their agreement level was measured with the kappa statistic, leading to the refinement of the inclusion and exclusion criteria.

A threat to data validity in this SLM is the data extraction bias, addressed through a pilot data extraction. The authors reviewed and discussed a pilot data extraction sample to improve the data extraction form. Another threat is the bias of classification schema. To avoid it, we relied on previous existing classifications when possible, such as the research strategies framework of Storey et al. [142]. We used the process groups from PMBOK [163] for the phases and familiar stakeholders' roles regarding the factors. We aggregated similar findings under labels that reflected the articles' original texts for naming the factors affecting software estimates. The authors held meetings for reviewing the factors and the categories in the SEXTAMT, and the types of effects of each factor.

As for research validity, there is the threat of lack of repeatability. One of the mitigation actions for this threat was involving more than one researcher during the process. Another action is to make all the SLM data publicly available, including decisions about inclusion and exclusion of papers, extracted data from primary studies, among others. Finally, we developed a research protocol to ensure replications or updates to this SLM. The protocol we developed and the discussions among the researchers involved helped mitigate the research method bias, another threat to research validity.

8. Conclusion

In this article, we presented an SLM about factors affecting expert judgment software estimates. We present such factors by three dimensions: the project phase they are likely to happen or to cause an impact over the estimates; the stakeholder that is responsible for a task or process to which the factor is linked, that directly causes the factor or that is directly impacted by the effects of the factor; and type of effect the factor causes. Some factors can have a negative effect, leading to errors when they are present, while others may have a positive or neutral effect. Such dimensions allow for easier navigation through the myriad of factors we found.

Most of the factors clustered at the planning and executing phases. It is natural to have factors at the planning phase, because estimating occurs primarily during such stage. At the executing phase, factors emerge because the dynamics of projects impact estimating error and accuracy. Moreover, most of the studies employed a research strategy of laboratory experiments, investigating one factor in a controlled setting with an experimental and control group. Also, they evaluated the difference of estimates between these groups to assess the impact of the factors.

Top factors — those that emerged in a higher number of studies — revealed the importance of issues beyond the estimation process. It is also necessary to improve the requirements engineering process, to deal with political issues, to consider the product characteristics, among others. Researchers have investigated a wide and varied set of factors. Therefore, we created a map to support readers in navigation through them: the SEXTAMT. If an interested reader desires to identify all factors that affect only one project phase, we provide them a classification through this dimension. If the reader desires to identify all factors given one stakeholder, we also provide this. Finally, if the reader wants to find out a class of factors given a specific effect — for instance, all factors that lead to improved accuracy — our map also has a dimension regarding this.

Our research confirms and aggregates existing results about factors affecting expert judgment estimates, a relevant contribution to move knowledge forward, especially when we organize such knowledge to facilitate understanding and future uses (for both research and practice). Also, the classification of measurement strategies is an additional relevant contribution. This enabled us to spot that our research community is missing the benefits of investigating more of noise as component of error.

Therefore, the dimensional map, facilitating the navigation through them, is a valuable research contribution. It can have many valuable uses in practice and software practitioners can employ the SEXTAMT factors as part of many different initiatives, such as:

- Diagnosing improvement opportunities to their estimation processes through the investigation of the most relevant factors in their contexts, considering their types of effects;
- building checklists to support estimators, considering especially the value adjustment characteristics;
- adapting checklists to aid debiasing interventions, considering especially the factors from the bias and estimation process categories;
- analyzing project risks by identifying the factors leading to larger risks of estimating error in their contexts and, therefore, of leading to failures to meet their commitments.

As for future work, we need to keep the SEXTAMT updated. Special care is due to the factors coming from unique articles: more investigation about them is needed. However, some philtering to identify the best candidates for more assessment is also necessary. Another critical issue is investigating the relationships among the factors to enrich the map with relevant mediation and moderation connections. A more complex framework can be helpful to identify the factors more likely to cause a more considerable impact over the estimates, focusing on them to adopt more cost-effective interventions during software improvement initiatives regarding estimation processes.

We highlight that another research issue comes from the software project dynamics that allows practitioners to adjust their work to fit an estimate when they need to, creating a "moving target" problem. This makes it harder to measure error and accuracy correctly. It also makes it harder to understand whether a factor contributed effectively to improving estimation accuracy or whether a software team just took advantage of higher project flexibility to create an illusion of accuracy. The solution to this comes from the judgment literature: measuring noise — unwanted variability from judgments that ideally should be identical [88]. Nevertheless, few studies from our SLM discuss this issue, revealing that our research community can benefit from understanding and using more of this concept.

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] A. Altaleb, M. Altherwi, and A. Gravell. 2020. An Industrial Investigation into Effort Estimation Predictors for Mobile App Development in Agile Processes. In *2020 9th International Conference on Industrial Technology and Management (ICITM)*, 291–296. DOI:https://doi.org/10.1109/ICITM48982.2020.9080362

[2] Abdullah Altaleb, Muna Altherwi, and Andy Gravell. 2020. A Pair Estimation Technique of Effort Estimation in Mobile App Development for Agile Process. In *Proceedings of the 2020 The 3rd International Conference on Information Science and System*. Retrieved November 26, 2020 from https://dl.acm.org/doi/abs/10.1145/3388176.3388212

[3] Abdullah Altaleb and Andrew Gravell. 2019. An Empirical Investigation of Effort Estimation in Mobile Apps Using Agile Development Process. *Journal of Software* 14, 8 (2019), 356–369.

[4] 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, (February 2019), 201–230. DOI:https://doi.org/10.1016/j.infsof.2018.10.006

[5] 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. DOI:https://doi.org/10.1145/1081706.1081761

[6] H. H. Arifin, J. Daengdej, and N. T. Khanh. 2017. An Empirical Study of Effort-Size and Effort-Time in Expert-Based Estimations. In *2017 8th International Workshop on Empirical Software Engineering in Practice (IWESEP)*, 35–40. DOI:https://doi.org/10.1109/IWESEP.2017.21

[7] 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*, Hong Kong, 6.

[8] Muesluem Atas, Stefan Reiterer, Alexander Felfernig, Thi Ngoc Trang Tran, and Martin Stettinger. 2018. Polarization Effects in Group Decisions. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization* (UMAP '18), Association for Computing Machinery, New York, NY, USA, 305–310. DOI:https://doi.org/10.1145/3213586.3225242

[9] Dirk Basten and Werner Mellis. 2011. A Current Assessment of Software Development Effort Estimation. In *2011 International Symposium on Empirical Software Engineering and Measurement*, IEEE, Banff, AB, Canada, 235–244. DOI:https://doi.org/10.1109/ESEM.2011.32

[10] 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), 4. DOI:https://doi.org/10.17705/1CAIS.03404

[11] Nick Benschop, Cokky A. R. Hilhorst, Arno L. P. Nuijten, and Mark Keil. 2020. Detection of early warning signals for overruns in IS projects: linguistic analysis of business case language. *European Journal of Information Systems* 29, 2 (March 2020), 190–202. DOI:https://doi.org/10.1080/0960085X.2020.1742587

[12] François Bergeron and Jean-Yves St-Arnaud. 1992. Estimation of information systems development efforts: A pilot study. *Information & Management* 22, 4 (April 1992), 239–254. DOI:https://doi.org/10.1016/0378-7206(92)90026-C

[13] P. Bhatt, G. Shroff, W. K, and A. K. Misra. 2006. An Empirical Study of Factors and their Relationships in Outsourced Software Maintenance. In *2006 13th Asia Pacific Software Engineering Conference (APSEC'06)*, 301–308. DOI:https://doi.org/10.1109/APSEC.2006.21

[14] Barry W. Boehm. 1984. Software Engineering Economics. *IEEE Transactions on Software Engineering* SE-10, 1 (January 1984), 4–21. DOI:https://doi.org/10.1109/TSE.1984.5010193

[15] Gary Boetticher and Nazim Lokhandwala. 2007. Assessing the Reliability of a Human Estimator. USA. Retrieved January 7, 2020 from https://dl.acm.org/doi/pdf/10.1109/PROMISE.2007.2

[16] Lars Bratthall, Erik Arisholm, and Magne Jørgensen. 2001. Program Understanding Behavior During Estimation of Enhancement Effort on Small Java Programs | SpringerLink. Retrieved February 11, 2021 from https://link.springer.com/chapter/10.1007/3-540-44813-6_30

[17] 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*, IEEE, 38–45. DOI:https://doi.org/10.1109/ICGSE.2015.10

[18] M. Brown, Henry Dirska, M. Pelosi, and M. Assadullah. 2013. Agile Method Software Development Estimation Biases. *International Journal of Advanced Research in Computer Science and Software Engineering* 3, 10 (2013).

[19] S. Bukhari and A. A. Malik. 2012. Determining the Factors Affecting the Accuracy of Effort Estimates for Different Application and Task Types. In *2012 10th International Conference on Frontiers of Information Technology*, 41–45. DOI:https://doi.org/10.1109/FIT.2012.16

[20] Lan Cao. 2008. Estimating Agile Software Project Effort: An Empirical Study. 11.

[21] Marco Conoscenti, Veronika Besner, Antonio Vetrò, and Daniel Méndez Fernández. 2019. Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development. *Journal of Systems and Software* 156, (October 2019), 126–135. DOI:https://doi.org/10.1016/j.jss.2019.06.075

[22] JS Davis. 1989. Investigation of predictors of failures and debugging effort for large MIS. *Information and Software Technology* 31, 4 (May 1989), 170–174. DOI:https://doi.org/10.1016/0950-5849(89)90034-7

[23] T. Dyba, T. Dingsoyr, and G. K. Hanssen. 2007. Applying Systematic Reviews to Diverse Study Types: An Experience Report. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, 225–234. DOI:https://doi.org/10.1109/ESEM.2007.59

[24] Kristian Marius Furulund and Kjetil Molkken-stvold. 2007. Increasing Software Effort Estimation Accuracy Using Experience Data, Estimation Models and Checklists. In *Seventh International Conference on Quality Software (QSIC 2007)*, 342–347. DOI:https://doi.org/10.1109/QSIC.2007.4385518

[25] T. J. Gandomani, H. Faraji, and M. Radnejad. 2019. Planning Poker in cost estimation in Agile methods: Averaging Vs. Consensus. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 066–071. DOI:https://doi.org/10.1109/KBEI.2019.8734960

[26] R. L. Glass, J. Rost, and M. S. Matook. 2008. Lying on Software Projects. *IEEE Software* 25, 6 (November 2008), 90–95. DOI:https://doi.org/10.1109/MS.2008.150

[27] S. Grapenthin, M. Book, T. Richter, and V. Gruhn. 2016. Supporting Feature Estimation with Risk and Effort Annotations. In *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 17–24. DOI:https://doi.org/10.1109/SEAA.2016.24

[28] A. R. Gray, S. G. MacDonell, and M. J. Shepperd. 1999. Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgment. In *Proceedings Sixth International Software Metrics Symposium (Cat. No.PR00403)*, 216–227. DOI:https://doi.org/10.1109/METRIC.1999.809743

[29] L. Gren, R. B. Svensson, and M. Unterkalmsteiner. 2017. Is It Possible to Disregard Obsolete Requirements? An Initial Experiment on a Potentially New Bias in Software Effort Estimation. In *2017 IEEE/ACM 10th International Workshop on Cooperative and*

Human Aspects of Software Engineering (CHASE), 56–61. DOI:https://doi.org/10.1109/CHASE.2017.10

[30] S. Grimstad and M. Jorgensen. 2007. The Impact of Irrelevant Information on Estimates of Software Development Effort. In *2007 Australian Software Engineering Conference (ASWEC'07)*, 359–368. DOI:https://doi.org/10.1109/ASWEC.2007.48

[31] S. Grimstad and M. Jorgensen. 2009. Preliminary study of sequence effects in judgment-based software development work-effort estimation. *IET Software* 3, 5 (October 2009), 435–441. DOI:https://doi.org/10.1049/iet-sen.2008.0110

[32] S. Grimstad, M. Jorgensen, and K. Molokken-Ostvold. 2005. The clients' impact on effort estimation accuracy in software development projects. In *11th IEEE International Software Metrics Symposium (METRICS'05)*, 10 pp. – 10. DOI:https://doi.org/10.1109/METRICS.2005.30

[33] Stein Grimstad and Magne Jørgensen. 2007. Inconsistency of expert judgment-based estimates of software development effort. *Journal of Systems and Software* 80, 11 (November 2007), 1770–1777. DOI:https://doi.org/10.1016/j.jss.2007.03.001

[34] Tanja M. Gruschke and Magne Jørgensen. 2008. The role of outcome feedback in improving the uncertainty assessment of software development effort estimates. *ACM Trans. Softw. Eng. Methodol.* 17, 4 (August 2008), 20:1-20:35. DOI:https://doi.org/10.1145/13487689.13487693

[35] Torleif Halkjelsvik and Magne Jorgensen. 2011. To read two pages, I need 5 minutes, but give me 5 minutes and I will read four: how to change productivity estimates by inverting the question. *Applied Cognitive Psychology* 25, 2 (2011), 314–323.

[36] Torleif Halkjelsvik and Magne Jørgensen. 2012. From origami to software development: A review of studies on judgment-based predictions of performance time. *Psychological Bulletin* 138, 2 (2012), 238–271. DOI:https://doi.org/10.1037/a0025996

[37] Torleif Halkjelsvik and Magne Jørgensen. 2018. How We Predict Time Usage. In *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*, Torleif Halkjelsvik and Magne Jørgensen (eds.). Springer International Publishing, Cham, 5–11. DOI:https://doi.org/10.1007/978-3-319-74953-2_2

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

[39] Susanne Halstead, Rosario Ortiz, Mario Córdova, and Miguel Seguí. 2012. The impact of lack in domain or technology experience on the accuracy of expert effort estimates in software projects. In *Proceedings of the 13th international conference on Product-Focused Software Process Improvement* (PROFES'12), Springer-Verlag, Berlin, Heidelberg, 248–259. DOI:https://doi.org/10.1007/978-3-642-31063-8_19

[40] N. C. Haugen. 2006. An empirical study of using planning poker for user story estimation. In *AGILE 2006 (AGILE'06)*, 9 pp. – 34. DOI:https://doi.org/10.1109/AGILE.2006.16

[41] Mei He, He Zhang, Ye Yang, Qing Wang, and Mingshu Li. 2010. Understanding the Influential Factors to Development Effort in Chinese Software Industry. In *Product-Focused Software Process Improvement* (Lecture Notes in Computer Science), Springer, Berlin, Heidelberg, 306–320. DOI:https://doi.org/10.1007/978-3-642-13792-1_24

[42] Raymond M. Henry, Gordon E. McCray, Russell L. Purvis, and Tom L. Roberts. 2007. Exploiting organizational knowledge in developing IS project cost and schedule estimates: An empirical study. *Information & Management* 44, 6 (September 2007), 598–612. DOI:https://doi.org/10.1016/j.im.2007.06.002

[43] J Hill, L. C Thomas, and D. E Allen. 2000. Experts' estimates of task durations in software development projects. *International Journal of Project Management* 18, 1 (February 2000), 13–21. DOI:https://doi.org/10.1016/S0263-7863(98)00062-3

[44] M. Host and C. Wohlin. 1998. An experimental study of individual subjective effort estimations and combinations of the estimates. In *Proceedings of the 20th International Conference on Software Engineering*, 332–339. DOI:https://doi.org/10.1109/ICSE.1998.671386

[45] J. Huang, H. Sun, and Y. Li. 2015. An empirical study of the impact of project factors on software economics. In *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 43–47. DOI:https://doi.org/10.1109/IEEM.2015.7385605

[46] Chrisholm Hugh. 1911. *The Encyclopaedia Britannica - Dictionary of arts, sciences, literature and general information* (11th ed.).

[47] M. J⊘rgensen. 2015. The effect of the time unit on software development effort estimates. In *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 1–5. DOI:https://doi.org/10.1109/SKIMA.2015.7399992

[48] Ali Javed, Ullah M. A., and Rehman Aziz-Ur. 2013. Factors Affecting Software Cost Estimation in Developing Countries. *International Journal of Information Technology and Computer Science* 5, (2013), 54–59.

[49] 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. DOI:https://doi.org/10.1109/TSE.2004.1274041

[50] M. Jørgensen. 2013. Relative Estimation of Software Development Effort: It Matters with What and How You Compare. *IEEE Software* 30, 2 (March 2013), 74–79. DOI:https://doi.org/10.1109/MS.2012.70

[51] M. Jørgensen. 2014. Fallacies and Biases when Adding Effort Estimates. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, 277–284. DOI:https://doi.org/10.1109/SEAA.2014.16

[52] 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. DOI:https://doi.org/10.1109/IWESEP.2018.00012

[53] M. Jorgensen, G. R. Bergersen, and K. Liestol. 2020. Relations Between Effort Estimates, Skill Indicators, and Measured Programming Skill. *IEEE Transactions on Software Engineering* (2020), 1–1. DOI:https://doi.org/10.1109/TSE.2020.2973638

[54] M. Jorgensen and S. Grimstad. 2005. Over-optimism in software development projects: “the winner’s curse.” In *15th International Conference on Electronics, Communications and Computers (CONIELECOMP’05)*, 280–285. DOI:https://doi.org/10.1109/CONIEL.2005.58

[55] M. Jørgensen and S. Grimstad. 2011. The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment. *IEEE Transactions on Software Engineering* 37, 5 (September 2011), 695–707. DOI:https://doi.org/10.1109/TSE.2010.78

[56] M. Jorgensen and S. Grimstad. 2012. Software Development Estimation Biases: The Role of Interdependence. *IEEE Transactions on Software Engineering* 38, 3 (May 2012), 677–693. DOI:https://doi.org/10.1109/TSE.2011.40

[57] M. Jorgensen and K. Molokken-Ostvold. 2004. Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method. *IEEE Transactions on Software Engineering* 30, 12 (December 2004), 993–1007. DOI:https://doi.org/10.1109/TSE.2004.103

[58] Magne Jørgensen. 2007. Individual Differences in How Much People Are Affected by Irrelevant and Misleading Information. Hellenic Cognitive Science Society, 347–352.

[59] Magne Jørgensen. 2007. A Critique of How We Measure and Interpret the Accuracy of Software Development Effort Estimation. In *First international workshop on software productivity analysis and cost estimation*, Japan.

[60] Magne Jørgensen. 2007. Forecasting of software development work effort: Evidence on expert judgement and formal models. *International Journal of Forecasting* 23, 3 (July 2007), 449–462. DOI:https://doi.org/10.1016/j.ijforecast.2007.05.008

[61] Magne Jørgensen. 2010. Selection of strategies in judgment-based effort estimation - ScienceDirect. *Journal of Systems and Software* 83, 6 (2010), 1039–1050.

[62] 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. DOI:https://doi.org/10.1016/j.infsof.2009.12.002

[63] Magne Jørgensen. 2011. Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation. *Information and Software Technology* 53, 12 (December 2011), 1382–1390. DOI:https://doi.org/10.1016/j.infsof.2011.07.001

[64] Magne Jørgensen. 2013. The influence of selection bias on effort overruns in software development projects. *Information and Software Technology* 55, 9 (September 2013), 1640–1650. DOI:https://doi.org/10.1016/j.infsof.2013.03.001

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

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

[67] 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. DOI:https://doi.org/10.1145/2601248.2601262

[68] 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), Springer International Publishing, Cham, 156–168. DOI:https://doi.org/10.1007/978-3-319-27033-3_11

[69] Magne Jørgensen. 2016. Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays. *Journal of Systems and Software* 117, (July 2016), 274–281. DOI:https://doi.org/10.1016/j.jss.2016.03.048

[70] Magne Jorgensen and Gunnar J. Carelius. 2004. An Empirical Study of Software Project Bidding. *IEEE Trans. Softw. Eng.* 30, 12 (December 2004), 953–969. DOI:https://doi.org/10.1109/TSE.2004.92

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

[72] Magne Jørgensen and Stein Grimstad. 2008. Avoiding Irrelevant and Misleading Information When Estimating Development Effort. *IEEE Software* 25, 3 (May 2008), 78–83. DOI:https://doi.org/10.1109/MS.2008.57

[73] 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. DOI:https://doi.org/10.1109/TSE.2009.2

[74] Magne Jørgensen and Torleif Halkjelsvik. 2010. The effects of request formats on judgment-based effort estimation. *Journal of Systems and Software* 83, 1 (January 2010), 29–36. DOI:https://doi.org/10.1016/j.jss.2009.03.076

[75] Magne Jørgensen and Torleif Halkjelsvik. 2020. Sequence effects in the estimation of software development effort. *Journal of Systems and Software* 159, (January 2020), 110448. DOI:https://doi.org/10.1016/j.jss.2019.110448

[76] Magne Jørgensen and Erik Løhre. 2012. First Impressions in Software Development Effort Estimation: Easy to Create and Difficult to Neutralize. IET, 216–222.

[77] 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. DOI:https://doi.org/10.1145/568760.568833

[78] 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), Springer, Berlin, Heidelberg, 174–184. DOI:https://doi.org/10.1007/978-3-540-24659-6_13

[79] Magne Jorgensen and Martin Shepperd. 2007. A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering* 33, 1 (January 2007), 33–53. DOI:https://doi.org/10.1109/TSE.2007.256943

[80] Magne Jørgensen and Dag I K Sjøberg. 2001. Impact of effort estimates on software project work. *Information and Software Technology* 43, (2001), 10.

[81] Magne Jørgensen and Dag I. K. Sjøberg. 2004. The impact of customer expectation on software development effort estimates. *International Journal of Project Management* 22, 4 (May 2004), 317–325. DOI:https://doi.org/10.1016/S0263-7863(03)00085-1

[82] 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 (February 2004), 79–93. DOI:https://doi.org/10.1016/S0164-1212(02)00160-7

[83] 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.

[84] Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. 2021. *Noise: A Flaw in Human Judgment*. Little, Brown Spark, New York.

[85] Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. 2021. Crime and noisy punishment. In *Noise: A Flaw in Human Judgment*. Little, Brown Spark, New York, 18–25.

[86] Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. 2021. Sequencing information in foresinc science. In *Noise: A Flaw in Human Judgment*. Little, Brown Spark, New York, 210–220.

[87] Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. 2021. Debiasing and decision hygiene. In *Noise: A Flaw in Human Judgment*. Little, Brown Spark, New York, 203–209.

[88] Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. 2021. Introduction: two kinds of error. In *Noise: A Flaw in Human Judgment*. Little, Brown Spark, New York, 9–15.

[89] Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. 2021. Your mind is a measuring instrument. In *Noise: A Flaw in Human Judgment*. Little, Brown Spark, New York, 39–41.

[90] H. Karna and S. Gotovac. 2014. Estimators characteristics and effort estimation of software projects. In *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, 26–35.

[91] Siobhan Keaveney and Kieran Conboy. Cost estimation in agile development projects. In *ECIS 2006 Proceedings*, 16.

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

[93] Stefan Koch and Gerhard Turk. 2011. Human Resource Related Problems in Agile and Traditional Software Project Process Models. *Int. J. Inf. Technol. Proj. Manag.* 2, 2 (April 2011), 1–13. DOI:https://doi.org/10.4018/jitpm.2011040101

[94] Robert Lagerström, Liv Marcks von Würtemberg, Hannes Holm, and Oscar Luczak. 2012. Identifying factors affecting software development cost and productivity. *Software Qual J* 20, 2 (June 2012), 395–417. DOI:https://doi.org/10.1007/s11219-011-9137-8

[95] Lucas Layman, Nachiappan Nagappan, Sam Guckenheimer, Jeff Beehler, and Andrew Begel. 2008. Mining software effort data: preliminary analysis of visual studio team system data. In *Proceedings of the 2008 international working conference on Mining software repositories* (MSR '08), Association for Computing Machinery, New York, NY, USA, 43–46. DOI:https://doi.org/10.1145/1370750.1370762

[96] A. L. Lederer and J. Prasad. 1998. A causal model for software cost estimating error. *IEEE Transactions on Software Engineering* 24, 2 (February 1998), 137–148. DOI:https://doi.org/10.1109/32.666827

[97] Albert L. Lederer and Jayesh Prasad. 1991. The validation of a political model of information systems development cost estimating. *ACM SIGCPR Computer Personnel* 13, 2 (August 1991), 47–57. DOI:https://doi.org/10.1145/122393.122398

[98] Albert L. Lederer and Jayesh Prasad. 1995. Perceptual congruence and information systems cost estimating. In *Proceedings of the 1995 ACM SIGCPR conference on Supporting teams, groups, and learning inside and outside the IS function reinventing IS* (SIGCPR '95), Association for Computing Machinery, Nashville, Tennessee, USA, 50–59. DOI:https://doi.org/10.1145/212490.212504

[99] Albert L. Lederer and Jayesh Prasad. 1995. Causes of inaccurate software development cost estimates. *Journal of Systems and Software* 31, 2 (November 1995), 125–134. DOI:https://doi.org/10.1016/0164-1212(94)00092-2

[100] Albert Lederer and Rajesh Mirani. 1990. Information System Cost Estimating: A Management Perspective. *Management Information Systems Quarterly* 14, 2 (December 1990). Retrieved from https://aisel.aisnet.org/misq/vol14/iss2/3

[101] Michael Lee, Marcus Rothenberger, and Ken Peffers. 2011. Identifying Effort Estimation Factors for Corrective Maintenance in Object-Oriented Systems. In *AMCIS 2011 Proceedings*. Retrieved from https://aisel.aisnet.org/amcis2011_submissions/186

[102] Valentina Lenarduzzi. 2015. Could social factors influence the effort software estimation? In *Proceedings of the 7th International Workshop on Social Software Engineering* (SSE 2015), Association for Computing Machinery, New York, NY, USA, 21–24. DOI:https://doi.org/10.1145/2804381.2804385

[103] M. R. Lind and J. M. Sulek. 1998. Undersizing software systems: third versus fourth generation software development. *Eur. J. Inf. Syst.* 7, 4 (December 1998), 261–268. DOI:https://doi.org/10.1038/sj.ejis.3000308

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

[105] Erik Løhre and Magne Jørgensen. 2016. Numerical anchors and their strong effects on software development effort estimates. *Journal of Systems and Software* 116, (June 2016), 49–56. DOI:https://doi.org/10.1016/j.jss.2015.03.015

[106] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. 2011. How social influence can undermine the wisdom of crowd effect. *PNAS* 108, 22 (May 2011), 9020–9025. DOI:https://doi.org/10.1073/pnas.1008636108

[107] 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. DOI:https://doi.org/10.1109/SEAA.2011.61

[108] A. Magazinius and R. B. Svensson. 2014. Effects of Feature Complexity on Software Effort Estimates – An Exploratory Study. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, 301–304. DOI:https://doi.org/10.1109/SEAA.2014.69

[109] 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 (August 2012), 1770–1781. DOI:https://doi.org/10.1016/j.jss.2012.03.026

[110] 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. DOI:https://doi.org/10.1145/1414004.1414042

[111] Viljan Mahnič and Tomaž Hovelja. 2012. On using planning poker for estimating user stories. *Journal of Systems and Software* 85, 9 (September 2012), 2086–2095. DOI:https://doi.org/10.1016/j.jss.2012.04.005

[112] Daniel Tadeu Martínez C. Branco, Edson Cesar Cunha de Oliveira, Leandro Galvão, Rafael Prikladnicki, and Tayana Conte. 2015. An Empirical Study About the Influence of Project Manager Personality in Software Project Effort. In *Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 2* (ICEIS 2015), SCITEPRESS - Science and Technology Publications, Lda, Portugal, 102–113. DOI:https://doi.org/10.5220/0005373001020113

[113] Olavo Matos, Luiz Fortaleza, Tayana Conte, and Emilia Mendes. 2013. Realising web effort estimation. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, Association for Computing Machinery, Porto de Galinhas, 12–23. DOI:https://doi.org/10.1145/2460999.2461002

[114] Patricia Matsubara, Bruno Gadelha, Igor Steinmacher, and Tayana Conte. 2021. Supplementary material for the SEXTAMT. *figshare*. DOI:https://doi.org/10.6084/m9.figshare.14502405.v1

[115] Steve McConnell. 2006. What is an “Estimate”? In *Software Estimation: Desmystifying the Black Art*. Microsoft Press, Redmond, 3–14.

[116] Steve McConnell. 2006. Flow of Software Estimates on a Well-Estimated Project. In *Software Estimation: Desmystifying the Black Art*. Microsoft Press, Redmond, 171–180.

[117] Steve McConnell. 2006. Introduction to Estimation Techniques. In *Software Estimation: Desmystifying the Black Art*. Microsoft Press, Redmond, 171–180.

[118] James McDonald. 2005. The Impact of Project Planning Team Experience on Software Project Cost Estimates. *Empir Software Eng* 10, 2 (April 2005), 219–234. DOI:https://doi.org/10.1007/s10664-004-6192-9

[119] F. McGarry, S. Burke, and B. Decker. 1998. Measuring the impacts individual process maturity attributes have on software products. In *Proceedings Fifth International Software Metrics Symposium. Metrics (Cat. No.98TB100262)*, 52–60. DOI:https://doi.org/10.1109/METRIC.1998.731226

[120] Emilia Mendes, Nile Mosley, and Steve Counsell. 2005. Investigating Web size metrics for early Web cost estimation. *Journal of Systems and Software* 77, 2 (August 2005), 157–172. DOI:https://doi.org/10.1016/j.jss.2004.08.034

[121] 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. DOI:https://doi.org/10.1109/ISESE.2003.1237981

[122] Kjetil Moløkken and Magne Jørgensen. 2005. Expert Estimation of Web-Development Projects: Are Software Professionals in Technical Roles More Optimistic Than Those in Non-Technical Roles? *Empirical Software Engineering* 10, 1 (January 2005), 7–30. DOI:https://doi.org/10.1023/B:EMSE.0000048321.46871.2e

[123] K. Molokken-Ostvold and K. M. Furulund. 2007. The Relationship between Customer Collaboration and Software Project Overruns. In *Agile 2007 (AGILE 2007)*, 72–83. DOI:https://doi.org/10.1109/AGILE.2007.57

[124] K. Molokken-Ostvold and M. Jorgensen. 2005. A comparison of software project overruns - flexible versus sequential development models. *IEEE Transactions on Software Engineering* 31, 9 (September 2005), 754–766. DOI:https://doi.org/10.1109/TSE.2005.96

[125] Kjetil Moløkken-Østvold, Nils Christian Haugen, and Hans Christian Benestad. 2008. Using planning poker for combining expert estimates in software projects. *Journal of Systems and Software* 81, 12 (December 2008), 2106–2117. DOI:https://doi.org/10.1016/j.jss.2008.03.058

[126] Kjetil Moløkken-Østvold and Magne Jørgensen. 2004. Group Processes in Software Effort Estimation. *Empirical Software Engineering* 9, 4 (December 2004), 315–334. DOI:https://doi.org/10.1023/B:EMSE.0000039882.39206.5a

[127] Ofer Morgenshtern, Tzvi Raz, and Dov Dvir. 2007. Factors affecting duration and effort estimation errors in software development projects. *Information and Software Technology* 49, 8 (August 2007), 827–837. DOI:https://doi.org/10.1016/j.infsof.2006.09.006

[128] Ariadi Nugroho and Christian F. J. Lange. 2008. On the Relation between Class-Count and Modeling Effort. In *Models in Software Engineering* (Lecture Notes in Computer Science), Springer, Berlin, Heidelberg, 93–104. DOI:https://doi.org/10.1007/978-3-540-69073-3_11

[129] M. C Ohlsson, C Wohlin, and B Regnell. 1998. A project effort estimation study. *Information and Software Technology* 40, 14 (December 1998), 831–839. DOI:https://doi.org/10.1016/S0950-5849(98)00097-4

[130] 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-. Retrieved December 19, 2019 from http://dl.acm.org/citation.cfm?id=942801.943632

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

[132] Jurka Rahikkala, Sami Hyrynsalmi, and Ville Leppänen. 2015. Accounting Testing in Software Cost Estimation: A Case Study of the Current Practice and Impacts. Tampere, Finland, 15.

[133] Jurka Rahikkala, Sami Hyrynsalmi, Ville Leppänen, and Ivan Porres. 2018. The Role of Organisational Phenomena in Software Cost Estimation: A Case Study of Supporting and Hindering Factors. *e-Informatica Software Engineering Journal* 12, 1 (2018), 167–198. DOI:https://doi.org/10.5277/e-Inf180107

[134] Jurka Rahikkala, Ville Leppänen, Jukka Ruohonen, and Johannes Holvitie. 2015. Top management support in software cost estimation: A study of attitudes and practice in Finland. *International Journal of Managing Projects in Business* 8, 3 (January 2015), 513–532. DOI:https://doi.org/10.1108/IJMPB-11-2014-0076

[135] Melvina Autar Ramessur and Soulakshmee Devi Nagowah. 2020. Factors Affecting Sprint Effort Estimation. In *Advanced Computing and Intelligent Engineering* (Advances in Intelligent Systems and Computing), Springer, Singapore, 507–518. DOI:https://doi.org/10.1007/978-981-15-1483-8_43

[136] Rianti Rozalina and Zulkefli Mansor. 2018. Validated Software Cost Estimation Factors for Government Projects using Rasch Measurement Model. *International Journal on Advanced Science, Engineering and Information Technology* 8, 5 (October 2018), 1890-1896–1896. DOI:https://doi.org/10.18517/ijaseit.8.5.6386

[137] 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, (November 2017), 1–21. DOI:https://doi.org/10.1016/j.infsof.2017.06.002

[138] 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. DOI:https://doi.org/10.1145/3167132.3167293

[139] Ofira Shmueli, Nava Pliskin, and Lior Fink. 2016. Can the outside-view approach improve planning decisions in software development projects? *Information Systems Journal* 26, 4 (2016), 395–418. DOI:https://doi.org/10.1111/isj.12091

[140] Thiago Silva-de-Souza and Guilherme Horta Travassos. 2017. Observing Effort Factors in the Test Design & Implementation Process of Web Services Projects. In *Proceedings of the 2nd Brazilian Symposium on Systematic and Automated Software Testing* (SAST), Association for Computing Machinery, New York, NY, USA, 1–10. DOI:https://doi.org/10.1145/3128473.3128480

[141] Klaas-Jan Stol and Brian Fitzgerald. 2018. The ABC of Software Engineering Research. *ACM Trans. Softw. Eng. Methodol.* 27, 3 (September 2018), 11:1-11:51. DOI:https://doi.org/10.1145/3241743

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

[143] Girish H. Subramanian, Parag C. Pendharkar, and Mary Wallace. 2006. An empirical study of the effect of complexity, platform, and program type on software development effort of business applications. *Empir Software Eng* 11, 4 (December 2006), 541–553. DOI:https://doi.org/10.1007/s10664-006-9023-3

[144] Girish Subramanian, Parag C. Pendharkar, and Dinesh Pai. 2017. An Examination of Determinants of Software Testing and Project Management Effort. *Journal of Computer Information Systems* 57, 2 (2017), 123–129.

[145] Safa Suliman and Gada Kadoda. 2017. Factors that influence software project cost and schedule estimation. Elnihood, Sudan. Retrieved February 10, 2021 from https://ieeexplore.ieee.org/document/8293053

[146] L. M. Taff, J. W. Borchering, and W. R. Hudgins. 1991. Estimeetings: development estimates and a front-end process for a large project. *IEEE Transactions on Software Engineering* 17, 8 (August 1991), 839–849. DOI:https://doi.org/10.1109/32.83918

[147] R. Tamrakar and M. Jørgensen. 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)*, 228–232. DOI:https://doi.org/10.1049/ic.2012.0030

[148] Binish Tanveer, Liliana Guzmán, and Ulf Martin Engel. 2017. Effort estimation in agile software development: Case study and improvement framework. *Journal of Software: Evolution and Process* 29, 11 (2017), e1862. DOI:https://doi.org/10.1002/smr.1862

[149] 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), Springer Berlin Heidelberg, 232–245.

[150] N. Tripathi, P. Seppänen, M. Oivo, J. Similä, and K. Liukkunen. 2017. The Effect of Competitor Interaction on Startup's Product Development. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 125–132. DOI:https://doi.org/10.1109/SEAA.2017.34

[151] Muhammad Usman, Jürgen Börstler, and Kai Petersen. 2017. An Effort Estimation Taxonomy for Agile Software Development. *Int. J. Soft. Eng. Knowl. Eng.* 27, 04 (May 2017), 641–674. DOI:https://doi.org/10.1142/S0218194017500243

[152] Muhammad Usman, Ricardo Britto, Lars-Ola Damm, and Jürgen Börstler. 2018. Effort estimation in large-scale software development: An industrial case study. *Information and Software Technology* 99, (July 2018), 21–40. DOI:https://doi.org/10.1016/j.infsof.2018.02.009

[153] 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. DOI:https://doi.org/10.1145/2745802.2745813

[154] 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, (December 2018), 286–309. DOI:https://doi.org/10.1016/j.jss.2018.09.054

[155] R. Valerdi. 2007. Cognitive Limits of Software Cost Estimation. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, 117–125. DOI:https://doi.org/10.1109/ESEM.2007.85

[156] Steven S. Vicinanza, Tridas Mukhopadhyay, and Michael J. Prietula. 1991. Software-Effort Estimation: An Exploratory Study of Expert Performance. *Info. Sys. Research* 2, 4 (December 1991), 243–262. DOI:https://doi.org/10.1287/isre.2.4.243

[157] S. Vijayakumar. 1997. Use of historical data In software cost estimation. *Computing Control Engineering Journal* 8, 3 (June 1997), 113–119. DOI:https://doi.org/10.1049/cce:19970303

[158] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (EASE '14), ACM, New York, NY, USA, 38:1-38:10. DOI:https://doi.org/10.1145/2601248.2601268

[159] Da Yang, Qing Wang, Mingshu Li, Ye Yang, Kai Ye, and Jing Du. 2008. A survey on software cost estimation in the chinese software industry. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and*

measurement - ESEM ’08, ACM Press, Kaiserslautern, Germany, 253. DOI:https://doi.org/10.1145/1414004.1414045

[160] A. H. Zapata and M. R. V. Chaudron. 2013. An empirical study into the accuracy of it estimations and its influencing factors. *Int. J. Soft. Eng. Knowl. Eng.* 23, 04 (May 2013), 409–432. DOI:https://doi.org/10.1142/S0218194013400081

[161] Ahmed Zarour and Samer Zein. 2019. Software Development Estimation Techniques in Industrial Contexts: An Exploratory Multiple Case-Study. *International Journal of Technology in Education and Science* 3, 2 (2019), 72–84.

[162] 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. DOI:https://doi.org/10.1016/j.infsof.2010.12.010

[163] 2017. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (6th ed.). Project Management Institute, Newtown Square, PA.

[164] 2017. ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary. *ISO/IEC/IEEE 24765:2017(E)* (August 2017), 1–541. DOI:https://doi.org/10.1109/IEEESTD.2017.8016712