PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

From a Gambit to a Defense with SwEDeL: an Approach to Defend Software Estimates from Pressure

PATRÍCIA GOMES FERNANDES MATSUBARA

Manaus
Dezembro, 2022

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

PATRÍCIA GOMES FERNANDES MATSUBARA

From a Gambit to a Defense with SwEDeL: an Approach to Defend Software Estimates from Pressure

Doctoral dissertation presented to the Informatics Graduate Program (*Programa de Pós-graduação em Informática*) – PPGI, at Universidade Federal do Amazonas (UFAM).

Advisor: Tayana Uchôa Conte, Ph.D.
Co-advisor: Bruno Freitas Gadelha, Ph.D.
Co-advisor: Igor Steinmacher, Ph.D.

Manaus
Dezembro, 2022

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

M434f

Matsubara, Patrícia Gomes Fernandes
From a gambit to a defense with SwEDeL : an approach to defend software estimates from pressure / Patrícia Gomes Fernandes Matsubara . 2022
213 f.: il. color; 31 cm.

Orientadora: Tayana Uchôa Conte
Coorientador: Bruno Freitas Gadelha
Coorientador: Igor Steinmacher
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Software effort estimation. 2. Negotiation. 3. Behavioral software engineering. 4. Pressure. I. Conte, Tayana Uchôa. II. Universidade Federal do Amazonas III. Título

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UFAM

FOLHA DE APROVAÇÃO

"From a Gambit to a Defense with SwEDeL: an Approach to Defend Software Estimates from Pressure"

PATRÍCIA GOMES FERNANDES MATSUBARA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Profa. Tayana Uchôa Conte - PRESIDENTE

Profa. Sabrina dos Santos Marczak - MEMBRO EXTERNO

Documento assinado digitalmente
gov.br EDUARDO SANTANA DE ALMEIDA
Data: 19/12/2022 08:20:21-0300
Verifique em https://verificador.iti.br

Prof. Eduardo Santana de Almeida - MEMBRO EXTERNO

Profa. Monalessa Perini Barcellos - MEMBRO EXTERNO

Documento assinado digitalmente
gov.br MARCELA SAVIA PICANCO PESSOA BASTOS
Data: 18/12/2022 23:59:32-0300
Verifique em https://verificador.iti.br

Profa. Marcela Sávia Picanço Pessoa Bastos - MEMBRO EXTERNO

Manaus, 15 de dezembro de 2022

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
MONALESSA PERINI BARCELLOS - SIAPE 1490891
Departamento de Informática - DI/CT
Em 19/12/2022 às 12:11

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/624699?tipoArquivo=O

To Takashi, Takeo, and Naomi

ACKNOWLEDGMENT

So that is it. The closure of my Ph.D. journey—and I survived it. Surprisingly, I even managed to enjoy it. The strongest feeling in this end is gratitude. I am especially thankful to God for His presence in every aspect of my life, including this achievement.

I thank my husband, Edson Takashi Matsubara, to whom I dedicate the "SEXTAMT". I hope we can navigate life together for many years to come, until the end of our lives and beyond. No words can express how much I love and admire you.

I thank my kids, Gabriel Takeo and Naomi Fernandes Matsubara, to whom I dedicate "Buying time for software development". I wish I could buy more time for us, so I could love and cuddle you as much as you deserve. You are a bless in my life and I love you with all my heart.

I thank my parents, Maria Aparecida de Jesus Fernandes and William Gomes Fernandes (in memoriam), representing all my relatives (including my in-laws) to whom I dedicate "Moving on from the software engineer's gambit". I know the sacrifices you made throughout your lives to make me who I am and to support my family.

I thank all my friends, to whom I dedicate "Trust yourself!". You played a prominent role in making me trust myself, encouraging and instigating me. I especially thank my friends from the Faculty of Computing at UFMS and my research group, USES.

Finally, I thank my advisors and the examination board, to whom I dedicate "The best defense is a good defense". I thank my advisor Prof. Tayana Conte (UFAM), and my co-advisors, Prof. Bruno Gadelha (UFAM) and Prof. Igor Steinmacher (NAU), for all their dedication and hard work to prepare me for my defense, my academic career, and my life. I thank the examination board for accepting our invitation to revise my work and allowing me to defend it. I am happy to see we are gathering people from all regions of Brazil. I thank Prof. Marcela Pessoa (UEA), Prof. Eduardo Souto (UFAM) and Prof. Raimundo Barreto (UFAM) from the North region; Prof. César França (UFRPE and Cesar School) and Prof. Eduardo Santana de Almeida (UFBA) from the Northeast region; Prof. Monalessa Perini Barcellos (UFES), Prof. Gleison Santos (UNIRIO), and Prof. Rodrigo Santos (UNIRIO), from the Southeast region; Prof. Sabrina Marczak (PUCRS), from the South region; and Prof. Edna Dias Canedo (UnB), from my birthplace and home: the beautiful Midwest of Brazil.

ABSTRACT

The estimation of software projects and tasks is a critical activity in software development and maintenance. Ultimately, people develop and maintain software to satisfy business goals. A problem arises when software estimates collide with such goals: *software practitioners deliberately change their estimates because of objectives outside the estimation context, yielding to pressure over their estimates, leading to product and life quality issues*. This reveals the behavioral side of software estimation: its results are affected by cognitive and social aspects, requiring more than technical skills to achieve success. Unfortunately, software professionals do not possess the skills needed to defend their estimates from pressure, even though they are the ones with enough technical knowledge to assess whether a business goal is feasible. Such situations lead to the establishment of unrealistic commitments. So, in this work our goal is *to provide support to estimators in defending their estimates and negotiating realistic commitments when they face pressure over their estimates*. We adopted a Design Science Research (DSR) approach to pursue it. We investigated the factors affecting expert judgment estimation through a Systematic Literature Mapping (SLM), filtering the ones closely related to deliberate changes of estimates, pressure, and the establishment of commitments. This allowed us to better understand these topics in the current research literature as part of the DSR relevance cycle. We also executed a qualitative study about the interaction of estimates and establishing commitments in the software industry to gain the practice's perspective as part of the DSR relevance cycle. We found evidence on how practitioners change their estimates to make them acceptable to other project stakeholders and use padding as a tool during the establishment of commitments, instead of defending their estimates and negotiating more realistic commitments. With the knowledge we gained from these studies and from a DSR rigor cycle focused on negotiation methods, we proposed an artifact as part of our DSR design cycle. The artifact, entitled SwEDeL (Software Estimates' Defense Lenses), is a set of lenses that embodies principles from negotiation to help estimators to change their passive posture of yielding to pressure to a more active attitude of engaging with other stakeholders, to better grasp their interests and needs, gain deeper understanding on how the estimate collides with business goals, and to look for alternatives to satisfy their customers and managers' interests without compromising their own. We assessed SwEDeL through a focus group and then created a digital simulation to present in a more dynamic format. The lenses and the digital simulation are a behavioral intervention to improve software practitioners' negotiation skills, empowering them to face pressure: a concrete step towards Behavioral Software Engineering. Finally, we assessed the digital simulation and SwEDeL in a controlled experiment with practitioners from the software industry. We collected data on participants' attitudes, subjective norms, perceived behavioral control, and intentions to perform the defense of their estimates in light of the Theory of Planning Behavior. Results show improved scores among experimental group participants after engaging with the digital simulation and learning about the lenses. They were also more inclined to choose a defense action when facing pressure scenarios than a control group exposed. Practitioners also perceived the set of lenses as useful in their current work environments. Collectively, these results show the effectiveness of the proposed approach and its perceived relevance for the industry.
Keywords: Software Effort Estimation, Negotiation, Behavioral Software Engineering.

FIGURE INDEX

TABLES INDEX

ABBREVIATIONS LIST

BATNA	Best Alternative to a Negotiated Agreement
BF_{+0}	Bayes Factor for null hypothesis for one-tailed tests
BF_{0+}	Bayes Factor for research hypothesis for one-tailed tests
BF_{01}	Bayes Factor for null hypothesis for two-tailed tests
BF_{10}	Bayes Factor for research hypothesis for two-tailed tests
BFI-20	Big-Five Intentory with 20 items
BHT	Bayesian Hypothesis Testing
BRE	Balanced Relative Error
BREBias	Balanced Relative Error - Bias
DSR	Design Science Research
MRE	Magnitude of Relative Error
MREBias	Magnitude of Relative Error - Bias
NHST	Null Hypothesis Significance Testing
OP	Observation Participant
PFG	Participant of the Focus Group
PMBOK	Project Management Body of Knowledge
Pred(X)	Percentage of Predictions with MRE within X
RAS	Rathus Assertiveness Scale
RQ	Research Question
SE	Software Engineering
SEPG	Software Engineering Process Group
SEXTAMT	Software Estimates of eXperTs: A Map of influencing facTors
SLM	Systematic Literature Mapping
SO	Secondary Objective
SQ	Secondary Question
SwEDeL	Software Estimates' Defense Lenses
TPB	Theory of Planned Behavior

TABLE OF CONTENTS

CHAPTER 1– INTRODUCTION

This chapter presents an introduction to this research. We contextualize our work in addition to presenting the motivation, research questions, goals, and the methodology we followed. Finally, we present the structure of this dissertation.

1.1. Context and Motivation

A chess opening defines sequences of moves and an overall plan that can impact the game until its end. A now-famous opening is the Queen's Gambit. An opening is called a gambit when one player sacrifices material (like a pawn) to gain compensation, such as to gain tempo or to create structural weaknesses on the opponent's side [1]. A game-theoretical perspective of software development evinces that software practitioners also play gambits as part of their daily practices. For instance, they can sacrifice the quality of the product to deliver software faster—a sort of software engineers' gambit that can lead to technical debt. Along these lines, Vidoni et al. [2] proposed a new perspective on technical debt management: an infinite game whose purpose is to continue playing indefinitely instead of winning, i.e., to make the software operational and used for as long as possible.

Considering software development from a game theory perspective, we can explore different strategies to improve our practice [3]. In this sense, software development is a game that unfolds in social settings, involving many different players [4]: organizations gathering people with varying roles working together, and operating in a market with clients, users, and potential competitors. In such a social context, software effort estimation poses more than just a technical challenge. For instance, pressure is a factor affecting software estimates [5]—something unexpected if we consider estimating as a technical prediction task only. Yet, it can lead to the complete rejection of conservative (and accurate) estimates [6] and to arbitrary changes to estimates [7] to make them acceptable to the expectations of other project stakeholders [8], especially when stakeholders suppose developers are not being as productive as they can be [9]. So, software professionals end up committing to a set of features associated with a schedule that their realistic estimates do not support [10]. This shows the behavioral side of software estimation, meaning we need more than technical skills to estimate tasks and projects correctly, but also soft skills to deal with pressure.

Moreover, poor estimation due to business motivations for earlier deadlines can lead to time pressure in software development [11]. Although time pressure can increase motivation and efficiency [12], it negatively affects software practitioners' quality of life, leading to emotions such as sadness and stress [13]. Previous research indicates that the consequences of negative emotions like these include decreasing developers' productivity and increasing delays in executing activities [14], which can put even more pressure on schedules: a hard-to-stop vicious cycle. Time pressure also negatively affects the product quality, making individuals take shortcuts during development, leading to minimal quality assurance tasks, and acting as an obstacle to reviews, among many other quality-related effects [11]. In the end, any productivity gains due to pressure probably are insufficient to compensate for the quality costs. It seems the software engineers' gambit is not working.

Returning to our game theory perspective, all this research shows the results of the strategy of yielding to pressure over software estimates. In summary, our research problem proposition is: *software practitioners deliberately change their estimates because of objectives outside the estimation context, yielding to pressure over their estimates during the establishment of commitments, leading to product and life quality issues*.

Making another comparison with chess, this problem proposition show that software engineers are like the player with the black pieces, because black possess a disadvantage at the start of the game that is also related to pressure. This disadvantage is that the white player has the power to make more effective threats, thus dictating the course of the game[1] [15]. Probably because of this, an opening is called a defense when its defining move is initiated by the black player, while it is called an attack when the defining move is made by the white player [15].

Therefore, in this dissertation we propose that software engineers also adopt a defense as a strategy to deal with pressure over estimates. Prominent software engineers have long ago proposed the exploration of this strategy in practice. Notably, Jones [6] emphasized that a relevant reason for unrealistic schedules and time pressure is the inability of software practitioners to defend their estimates. Brooks [16] called out software engineers "to stiffen their backbones and defend their estimates", and McConnell [17] also stressed the need for such a strategy. Thus, the research question we aim to answer through our research is:

Research Question: How can we support software practitioners in defending their estimates and negotiating realistic commitments?

When discussing the strategy of defending estimates, McConnell [17] proposed the use of the principled-negotiation method [18]. Although he provided experienced advice in the

[1] A power known as the "initiative".

form of tips on the use of principled negotiation, he did not present a structured method completely adapted for the software estimation context. Additionally, the author did not empirically evaluate the tips. Also, one of the authors of the principled-negotiation method devised other supporting methods in the last years: the breakthrough strategy [19] and the positive no-method [20]. Thus, in this dissertation we further explore the strategy of defending software estimates instead of yielding to pressure, creating an artifact to support it [21].

In the next section, we present the research objectives that we defined for our research. We also present the research methodology in detail, together with the structure of this dissertation.

1.2. Research Objectives

Our primary objective in this research project is ***to provide support to estimators in defending their estimates and negotiating realistic commitments when they face pressure over their estimates***. In this context, an estimator is any person generating a software estimate (of size, effort, duration, or productivity) and communicating it or creating commitments based on it. To satisfy our primary objective, we devised the following secondary objectives:

- SO 1 - Build a body of knowledge of factors affecting expert-judgment software estimates to understand and position the pressure factor compared to others. We aim to provide an improved perspective of the problem considering the existing research literature.
- SO 2 - Investigate how software practitioners use estimates to establish software development commitments in the software industry to understand more of the effects in the field. We look to provide an improved perspective of the problem considering the software practice.
- SO 3 - Develop an artifact to support estimators to defend their estimates and negotiate realistic commitments when facing pressure.
- SO 4 - Assess the artifact to identify its capacity of improving the intentions of estimators in defending their estimates and negotiating realistic commitments when facing pressure.

1.3. Research Methodology

In this section, we present our research methodology, as described also by Matsubara [22]. At the beginning of our project, we envisioned the creation of an artifact to support estimators in their daily practice. Therefore, we chose the Design Science Research (DSR) methodology as it supports the creation of artifacts in context [23], enabling the creation of solutions for practical problems [24]. Figure 1.1, adapted from Hevner [25], illustrates the research steps (with letters from A to G). Each rounded gray rectangle represents one study that ultimately aims to answer our research question.

The first two studies in this research are literature reviews (A and B) that enabled us to refine the research scope and the foundations of our knowledge base, respectively. The preliminary literature review (A) aimed at identifying influence factors for software project estimates. We conducted a search for papers using snowballing procedures, starting with the work by Halkjelsvik and Jørgenssen [26], [27]. The preliminary review contributed to the understanding of our research problem. It also offered input to the planning of the SLM (C) conducted later on.

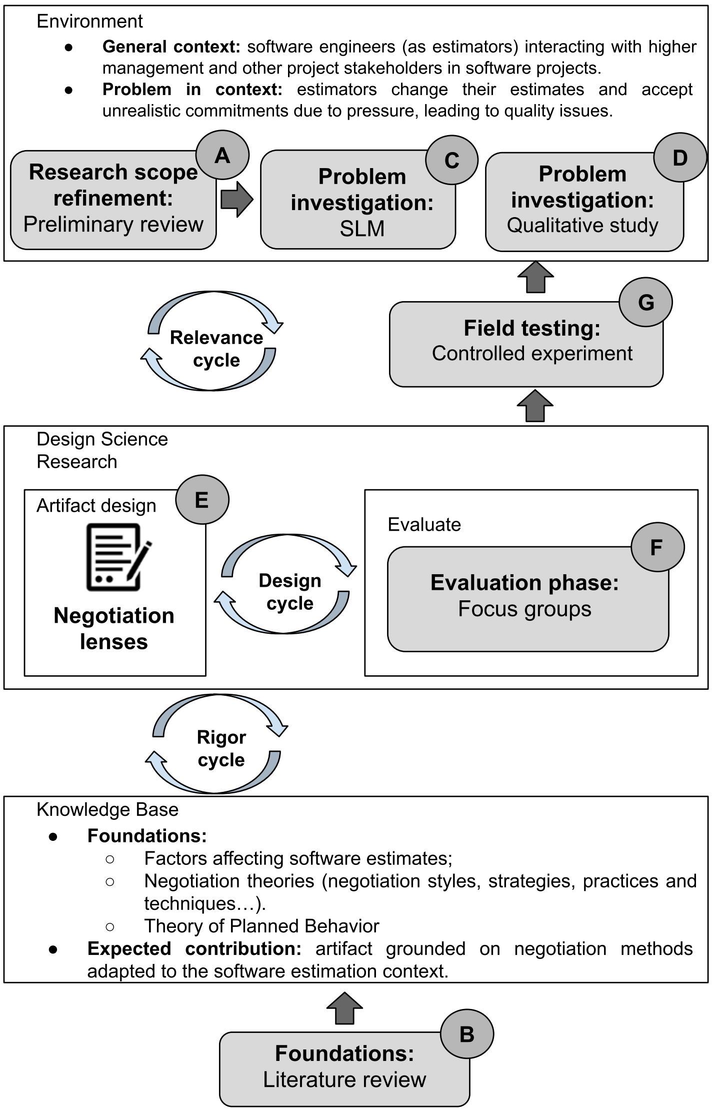


Figure 1.1 Research Methodology

In parallel, we investigated literature about negotiation theories (B), starting with the work of (i) Lande [28]; (ii) Roloff, Putnam, and Anastasiou [29]; and (iii) Brett and Thompson [30]. This helped us to learn about the topics investigated by negotiation theorists and the search terms to use in our exploratory search. After that, we searched for more reviews about negotiation at Google Scholar[2] and Scopus[3]. We also added to our knowledge base the principled-negotiation method proposed by Fisher, Ury, and Patton [18] given that McConnell [17] argue it is a feasible method to apply in the estimation context. We also added the breakthrough strategy [19] and the positive no-method [20] because they can be considered as continuing the work started with principled negotiation. As our research project evolved, we also added the Theory of Planned Behavior to aid us in evaluating our proposed artifact. We present the foundations of our knowledge base in CHAPTER 2.

The subsequent studies are a Systematic Literature Mapping (SLM - C) and a qualitative study (D). These studies aimed to better investigate the problem and gain more insight into the research project's environment. The SLM (C) focused on the factors affecting expert-judgment estimation [5] to satisfy the Specific Objective (SO 1). We report this study it in CHAPTER 3. We decided to focus on expert-judgment estimation because it is the most used method in the software industry [31], [32], [33], and on the rise as a research topic in software effort estimation [34].

The qualitative study (D) focused on how software practitioners in the software industry use the estimates when establishing commitments with their clients and higher management to satisfy SO 2. We collected data through observation sessions and interviews with practitioners in varying roles and from five different companies [8]. We describe the study in detail together with the results in CHAPTER 4.

The next activity was the artifact design (E), which is part of the design cycle and is related to SO 3. It resulted in the first version of our artifact, a set of lenses called SwEDel: Software Estimates' Defense Lenses. In this context, the idea of proposing the artifact in the format of lenses was inspired by the design lenses proposed to guide the design of gamified systems [35]. Design lenses encapsulate a design principle, and a set of focusing questions, supporting the designer to take a mental perspective regarding the design issue the lens focus on. The defense lenses share a similar purpose: supporting estimators on taking a different perspective when facing pressure over their estimates. The lenses were adapted to the

[2] https://scholar.google.com.br/
[3] https://www.scopus.com/home.uri

estimation context, considering pressure scenarios that estimators face. Moreover, they are based on the principled negotiation method [18], the breakthrough strategy [19], and the positive no method [20]. We provide the complete set of lenses and their rationale, based on the previously mentioned negotiation methods; see CHAPTER 5.

Following, still in the design cycle, is the execution of studies to evaluate the artifact and to satisfy the SO 3. The evaluation phase in this project included static validation, which involves presenting the method to industry practitioners to get feedback [36]. We carried this out as a study based on a focus groups [37] (F), which gave us feedback for the artifact design activity and preliminary evidence on the usefulness of the artifact [21]. We explain the focus groups and present its results in CHAPTER 6.

We used the results from the focus group (F) as input to another round of the design cycle. We considered that adopting a defensive strategy in real-world settings requires a behavior change of practitioners. This is a topic of study of behavioral and social sciences, areas with hundreds of years of studying human behavior. We can use their knowledge as leverage to better understand how SE practitioners behave, think, and feel [38], and to craft interventions based on such knowledge. Therefore, we enlisted behavior change interventions to our aid—a concrete step toward Behavioral Software Engineering, i.e., the study of cognitive, behavioral and social aspects of Software Engineering [38]. Boosts are a promising class of behavioral interventions: they improve competencies, enabling individuals to exercise their agency and empowering them to make better decisions [39]. Therefore, we incorporated to our artifact a boost intervention, in the form of a digital simulation: technology-based simulations that model a process or a system [40]. It is composed of lightweight interactive videos, with a few paused moments when participants choose one action to take in typical pressure scenarios. We present the digital simulation in CHAPTER 5.

After creating the defense lenses and designing the digital simulation to present them, we moved to field testing, with a controlled experiment (G) in the context of the relevance cycle of DSR. We executed a controlled experiment with industry practitioners, to understand their impact upon software practitioners' behaviors in their daily practices. More specifically, we examine whether participation in the digital simulation affects professionals' intentions to adopt the strategy we proposed, thus moving on from the software engineers' gambit. We considered that intentions are the immediate antecedent of behavior, as posited by the Theory of Planned Behavior (TPB) [41]. We also collected data on attitudes, subjective norms, and perceived behavioral control, as these are antecedents of intentions. With this, we satisfied the SO 4 described in Section 1.2. We explain the controlled experiment and its results in

CHAPTER 7. Next, we finish the dissertation with presenting our final considerations and summary of our contributions in CHAPTER 8.

1.4. Contributions

Regarding problem investigation, one of our contributions with this dissertation is a map of factors affecting expert-judgment software estimates resulting from our SLM, that is comprehensive enough to benefit other researchers investigating topics related to software effort estimation. From the qualitative study, our contribution is empirical evidence that software practitioners change their software software estimates to make them defensible and that they have varying reasons for padding software estimates in the software industry.

Moreover, a substantial contribution of our work are the artifacts we created: the defense lenses and the digital simulation. Also, our approach incorporates principles from negotiation, thus holding the potential to promote the learning of this soft skill among software practitioners. It also aims to provide means for estimators to deal with the behavior of stakeholders,—a step towards Behavioral Software Engineering, i.e., to the “study of cognitive, behavioral and social aspects of software engineering performed by individuals, groups or organizations”[38].

Regarding the evaluation of our artifacts, our contributions are the empirical evidence about the intentions of software practitioners to adopt the defense lenses and of the digital simulation’s and lenses’ perceived usefulness. Also, we incorporated to our analysis the Theory of Planned Behavior, a consolidated social science theory, exploring how it can be useful to understand behavior change in our field.

CHAPTER 2 – BACKGROUND

In this chapter, we detail the concepts of estimate and of negotiation. We also discuss the negotiation methods we used as the foundations of our proposed solution, the intersection of negotiation and estimates in Software Engineering, and the Theory of Planned Behavior.

2.1. Introduction

ISO/IEC/IEEE 12207 standard defines that one of the tasks of any project planning process is to "define and maintain a project schedule based on management and technical objectives and work estimates" [42]. This implies that plans are formulated both from estimates and objectives. Also, it means we craft our plans to achieve specific results, often in the form of targets: statements about desirable business outcomes [43]. After all, companies have their business goals, sometimes defined in the form of strategic plans, derived from their market analysis, business opportunities, and even potential threats and risks. However, a desirable or even mandatory target considering market issues is not necessarily achievable [43].

Unfortunately, sometimes targets can influence software estimates or be imposed on software teams and practitioners, leading to unrealistic commitments. Therefore, we continue this chapter in Section 2.2 by discussing software estimation—the software engineering activity that is the topic of this dissertation—and its interaction with related concepts, such as targets and commitments, relevant for understanding our research problem and proposal.

Considering the problem of yielding to the pressure that we described in Section 1.1, we propose the use of negotiation principles and methods to devise a supporting artifact for software practitioners to change the game and defend their estimates instead of yielding. Therefore, Section 2.3 introduces the definition of negotiation and discuss relevant concepts. Sections 2.4, 2.5, and 2.6 present the three negotiation methods that support our proposed approach. Next, Section 2.7 shifts a bit the content to discuss the Theory of Planned Behavior as a lens to help us to understand whether practitioners are likely to change their behavior from yielding to pressure to defending software estimates. Finally, Section 2.8 presents the intersection of negotiation and software estimates in the Software Engineering literature.

2.2. Plans, Estimates, Targets, and Commitments

When planning software projects, software estimates are the element used to define what is feasible and what is not. They are the predictions made about a project variable, like effort, cost, or duration [44]; a projection from the past to the future [45]. A major issue in software projects is that targets—statements about desirable business outcomes [43]—and estimates can collide, causing internal conflicts when defining a commitment [46]. One example is the scenario McConnell [43] presents, with an executive asking a project lead for an estimate of how long one project will take while also informing that he needs the software in three months (the target), without the possibility of including additional team members—and pressing for the team leader to commit with this target. Software professionals end up accepting unrealistic commitments because of this.

Therefore, technical staff needs to defend their estimates in the face of aggressive business targets, to reach realistic commitment with business staff and clients. McConnell [17] proposed using the principled-negotiation method [18] in the discussions of estimates, targets, commitments, and plans. In the following sections, we discuss the definition of negotiation, present some negotiation methods, and point to related work, positioning our research project in comparison to them.

2.3. Negotiation

Fisher, Ury, and Patton [18] define negotiation as a means of people getting what they want, through a back-and-forth communication process that aims at reaching agreement when the involved parties have opposed and/or shared interests. Ebner et al. [47] discuss the definition of negotiation through elements that make specific situations have negotiation-like characteristics:

- **The parties' perspective:** Did the parties prepare to negotiate? Do they consider the situation to be a negotiation?
- **Purpose:** Is each party trying to get something with the interaction? Are they trying to gain something or improve their situation?
- **Structure or relationship:** Do the parties need each other to achieve their purpose? What is their interdependence level? To what extent do the parties need to work together to achieve to their purpose, compared to the power each one has for making one-sided decisions?
- **Action:** Are the parties communicating back-and-forth? Are they questioning each other? Are they sharing interests, needs, or demands? Are they making offers or creating options?

- **Result:** Is there any explicit agreement or has a demand been accepted? Regardless of agreements, did the parties' behavior change?

We can benefit greatly from approaching situations that present the abovementioned characteristics with a negotiation mindset, employing negotiation skills to deal with them [47]. Therefore, this work adopts their negotiation definition, arguing that the interaction between the estimation process and the establishment of commitments lead to a "negotiation-ish" situation, and that we could take advantage of a negotiation mindset in this context.

The issue then shifts from whether the situation is a negotiation to how to negotiate. Two major negotiation models exist: distributive and integrative negotiation [28]. The first focuses on dividing scarce resources, while the second focuses on looking for mutually beneficially agreements [48]. The integrative model is also called principled negotiation [28]. Principled negotiation was proposed by Fisher, Ury, and Patton [18] in a seminal work that shifted the approach to negotiation from fixed positions and bargaining to a more flexible one, focused on the interests of the parties [49]. It also revolutionized the teaching of negotiation in many different fields [50], and impacted how practitioners think about negotiation [51].

Later, two other methods built upon principled negotiation, providing additional guidelines when people insist on uncooperative behaviors [19], or make unacceptable demands or requests [20]. Collectively, the principles and steps of such methods have impacted a wide variety of domains: from health care (e.g., pediatric operating rooms [52]), to personal improvement (e.g., recommendations for raising interpersonal assertiveness [53]), to military (e.g., practical guide for negotiating in the military [54]), and others. In the following sections, we summarize principled negotiation and negotiation methods that derived from it. We do not intend to present a complete summary but rather to provide an overview of the steps proposed in each method.

2.4. Principled Negotiation

Fisher, Ury, and Patton [18] proposed principled negotiation to get to yes in negotiations during the ‘80s. The method is based on four principles, which we detail in this section:

- Separate the people from the problem.
- Focus on interests, not positions.
- Invent options for mutual gains.
- Insist on using objective criteria.

The first principle, **separate the people from the problem**, means that one needs to recognize that one negotiates with people. Additionally, every negotiator has two interests: over the substance of negotiation and over their relationships with other parties. Positional bargaining leads to unnecessary conflicts among such interests, and negotiators may make substantive concessions to solve people problems. Fisher, Ury, and Patton [18] explain that one must care for differences in thinking, emerging feelings, and misinterpretations during communication as part of separating the people from the problem.

The second principle is to **focus on interests, not positions**. Focusing on positions—like the price offers people make when negotiating over a product or the deadlines they argue for when establishing a commitment in a software project—can create an impasse. Getting to know the other side's desires and concerns—their interests—helps to look for alternatives that meet both sides' interests, built upon shared and complementary interests. People have many different interests, and multiple positions may satisfy them.

The third principle is to **invent options for mutual gains**. The options should explore differences in parties' interests, the value placed on time, risk aversion, and others. This principle requires people to separate inventing from deciding, and brainstorming techniques are useful. It also requires a shift from the mindset of "solving their problem is their problem". Preparing multiple options equally acceptable to one's side and asking the other side to choose the one they prefer raises an agreement's chances. Options that include low-cost items to us but that greatly benefit the other side are precious for creating mutual gain. Another technique to generate multiple options is to think of "weaker" versions of one's preferred options if agreement gets difficult. Providing a convincing rationale and the consequence for each option also paves the way towards an agreement.

The fourth principle is to **insist on using objective criteria**. Even after adhering to the three first principles, interests may still conflict. Using objective criteria supports reaching an agreement based on principles instead of pressure. Objective criteria may come in the form of fair standards, like market value, precedent, efficiency, scientific judgment, and others. They may also come in the form of fair procedures, like taking turns, drawing lots, letting someone else decide or "one cuts, the other chooses".

2.5. The Breakthrough Strategy

After the publication of the principled negotiation method, people raised the issue of how to negotiate with difficult people or during complicated situations. In other words, how to get past no from other parties to reach an agreement? In this context, Ury [19] proposed the breakthrough strategy in five steps to overcome cooperation barriers:

- Don’t react: go to the balcony.
- Don’t argue: step to their side.
- Don’t reject: reframe.
- Don’t push: build them a golden bridge.
- Don’t escalate: use power to educate.

The first step is to **go to the balcony** to suspend one’s natural reactions—such as striking back, giving in, or breaking off—when the other side refuses to reach an agreement. None of them helps to move one’s interests forward truly. It might be the case that the other side provokes a reaction by employing different tactics during the negotiation. These tactics can include stone walls, that is, a refusal to budge; attacks, either to one’s proposals, credibility, or authority; or tricks that take advantage of one’s assumptions about their good faith, such as data manipulations, authority ploys, or last-minute demands. In contexts where such tactics are used, one needs to go to the balcony to reflect, buying time to gain perspective of the situation and naming their tactics to neutralize their effects.

Next, one needs to **step to their side**, instead of arguing in the face of their disagreement to disarm them. This involves active listening, inviting the other side to talk about everything they have to say about the matter, and focusing on what they say. Paraphrasing and asking for corrections are also part of this process to ensure one is truly hearing what they say. Another action is to acknowledge their point, including their feelings. If anyone on one’s side have done any wrong in the past, one must offer a sincere apology while still projecting confidence in oneself. Agreeing whatever one can without conceding is an additional recommendation. So, one must look for any opportunities to agree instead of focusing on disagreements. Say yes whenever possible also helps to reduce tension. Be sincere and watch one’s body language is also important. Acknowledging their authority and competence to dispel any feelings of threat they might have.

After hearing them, it is one’s turn to speak, and one must express one’s point of view without provoking them. This can be done with “I-statements”: talk about the impact of the problem on oneself, instead of talking about what they have done wrong. If talking about one’s perceptions about them, they can easily disagree, and one can also be wrong. However, it will be hard for them to challenge one’s experiences. Acknowledge any differences with optimism, expressing that one genuinely believes they can be resolved.

If they still hold firm to their position, one should not reject it. The third step is to **reframe**: redirect their attention from the positions to interests, creative options, and fair standards. So, take their position as a piece of useful information about what they want and investigate it. Ask problem-solving questions and let the problem be their teacher. For instance, one can ask why they want what they say they want; offer an additional option and ask why not; ask for their advice, getting them involved; or ask what makes their position fair.

When the other side use tactics, we can also reframe them. Ury [19] explains that if they present stone walls, adopting firm positions from which they will not budge, one can ignore the stone wall, reinterpret it as an aspiration, or take it seriously but also test it. If they attack us, one can ignore it or reframe it in many different ways like as an attack on the problem, as friendly, from past wrongs to future remedies, or from "you" and "me" to "we". If they use tricks, one can play along as if they were negotiating in good faith, while asking clarifying questions to check whether they are sincere. If caught in their contradiction, one can act confused instead of challenging them. One can also design reasonable requests that they will accept if they are genuinely cooperative.

If none of these works, and they keep using tricks, one can let them know that one knows what they are doing without accusing them. Talk about their tactics nicely and do not treat them as dirty tricks. If bringing their tactics up still does not work, negotiate about the negotiation. One can discuss the fairness of the tactics and make specific requests about what one wants to change.

The fourth step is to **build them a golden bridge**, to walk out from their position to the agreement one wants. They might still resist agreement for many different reasons like if it was not their idea, they still have unmet interests, fear of losing face, or feel the decision seems too big and taken too fast. One's job is to help them overcome these obstacles to agreement. One can involve the other side, asking for constructive criticism of our proposals or offering them choices to select from.

If they still reject all proposals, the fifth and final step is to **use power to educate**. Let the other side know the consequences of no agreement. One can do it by asking reality-testing questions or warning them—which is entirely different from threatening. Threats declare intentions to punish, while a warning is a notice of future danger resulting from the situation itself. If reality-testing questions and warnings do not work, demonstrate one's power through one's BATNA (Best Alternative to a Negotiated Agreement) [19]. If the other side still refuses to negotiate, deploy one's BATNA without provoking them. Much care is needed in this case because the use of power from one's side can lead them to use their power to fight back. So, the BATNA is one's last resort. Use it showing that one regrets doing so. Also, remind them of the golden bridge frequently, always giving them a choice to come back to the negotiation table. Clarify that one do not want to win over them: the goal is a mutually satisfactory and lasting agreement.

2.6. The Positive No Method

There are situations in which people must stand for their no because it is the best for their interests. Still, people can react poorly when facing a request or demand, falling in the three-A trap [20], that Figure 2.1 shows. The three-A trap involves either (i) accommodating, when people say yes when they really feel like saying no; (ii) attacking, when people say no poorly and damage the relationship; or (iii) avoiding, when people say nothing at all. To enable people to say no and still protect their relationships, Ury [20] proposed an alternative to the three-A trap: the Positive No method, structured as Figure 2.1 depicts.

A Positive No is composed of a first Yes, then No, and a second Yes. The **first Yes** expresses the interests of the one saying no and is internally focused. The **No** asserts their power. The **second Yes** aims at maintaining or improving the relationship with the person receiving the no. It is an invitation to an alternative agreement focused on satisfying the interests of everyone involved. A Positive No structure helps people create what they want, protect what they value, and change what no longer works [20]. The Positive No method has three stages that Figure 2.2 illustrates: (i) prepare, (ii) deliver, and (iii) follow through.

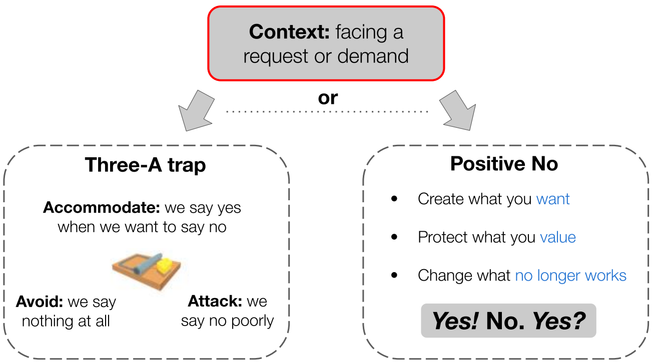


Figure 2.1 - The three-A trap and the Positive No.

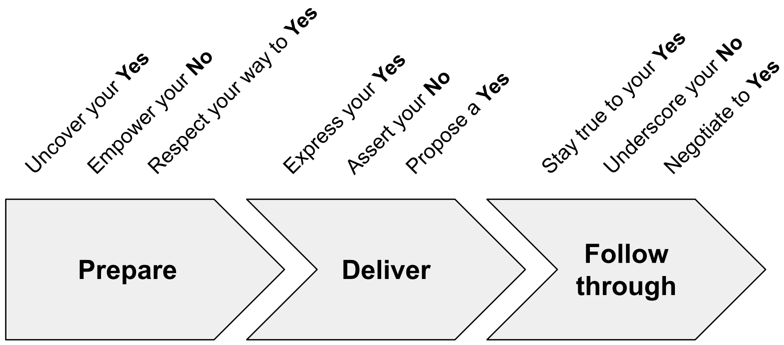


Figure 2.2 - Stages of the Positive No Method.

The first stage is to **prepare**, starting with uncovering the first Yes: the reasons why one wants to say No. It involves discovering one's interests, needs, and values. Then one crystallizes them in a single intention that reminds of one's commitment to oneself. Next, empower one's No, by devising a Plan B: a course of action one will take in the case of no agreement. This alternative plan must be independent of the other side, and it is the last resort—corresponding to one's BATNA. It gives the psychological freedom to stand for one's No. If one's Plan B is not attractive or feasible, one also must work on improving it. The last step to prepare one's Positive No is to respect one's way to Yes, by trying to understand the other side utterly. It involves listening to the other side attentively, asking clarifying questions about their demands and requests. It also involves acknowledging their point of view and letting them know one values them.

The second stage is to **deliver** one's No. It starts with expressing one's Yes, rooting the No in the power of positive intention, and clarifying one's motives for saying No. It requires sticking to the facts and describing one's feelings and interests. One may also resort to shared interests and shared standards with the other side. Then, let one's No flow from one's Yes, power, and respect for them. The final step is to propose a Yes. If one says No to a demand, offer a third option to reconcile one's and their interests. If one says No to a behavior, make a constructive, clear, and feasible request for a change.

The third stage is to **follow through**. If the other side rejects one's No, pause before responding. It is time to listen respectfully again: acknowledging their point of view without conceding one's. Show that one is listening: repeat what they say with one's words. If they attack, use the power of not reacting. Recognizing their tactics, if they use any, one can neutralize them. It is also time to underscore one's No, with positive power. If necessary, repeat one's No.

Additionally, let reality be their teacher about what will happen if they do not respect one’s interests, needs, and values. Ask reality-testing questions to reflect on the consequences of refusing one’s No. Like in the breakthrough strategy, warnings are also valid, but threats are not. Use the power to reflect on logical consequences that derive from their actions and the situation. If they keep resisting, it is time to deploy one’s Plan B to show that the warnings were not bluffs. The more powerful Plan B is, the more respect one needs to demonstrate when deploying it, to keep doors open to cooperation. Plan B can strain the relationship, so use it sparingly and positively: do not frame it as a punishment for their behavior but as a defense of one’s interests, which one would appreciate not having to use.

Lastly, one need to negotiate to Yes, looking for their interests that might still be unmet by one’s proposals. However, one should not compromise essentials for one’s side. Investigate whether they are rejecting one’s No because accepting it will not be approved by third parties they have to respond to. If that is the case, help them to gain approval from others. One needs to make them look good to other relevant people by accepting one’s No and any proposals one have made consequently. This is also essential to cultivate a healthy relationship because it is key to successfully implementing agreements.

Adopting the negotiation principles embodied in the methods discussed so far requires a behavior change from software practitioners. Therefore, in the next section we present the ideas of a social science theory that can help us in understanding whether software professionals are likely to do so, thus contributing to the knowledge base we used in one of our studies, which we describe in CHAPTER 7.

2.7. The Theory of Planned Behavior

Among the most used social science theories in Software Engineering research according to Lorey et al. [55], the Theory of Reasoned Action is the one that focuses on people's overt behaviors [56]. It has been revised and expanded, leading to the Theory of Planned Behavior (TPB) [41]. Therefore, we chose the TPB to understand more about the behavior of defending software estimates in our study. The TPB affirms the immediate antecedent of behavior is intention, which is a combination of attitudes, subjective norms, and perceived behavioral control regarding the behavior [57]. Kan and Fabrigar [56] define each of the theory's components:

- Behavior is one or more overt actions performed by a person, conceptualized in terms of actions, target, context, and time. It is the object of interest of the researcher.
- Intention is the person's perceived likelihood of performing the behavior. It is the immediate antecedent of the behavior.
- Attitude towards the behavior is a person's evaluation (favorably or not) of performing the behavior. It is an aggregate of behavioral beliefs: beliefs that the behavior leads to certain outcomes and how good or bad the individual evaluates such outcomes.
- Subjective norm regarding the behavior is a person's perception about how other important people consider the behavior. It is a function of normative beliefs the individual holds regarding what a referent thinks the individual should or should not do. It also includes the motivation the individual must comply with they believe each specific relevant referent thinks.
- Perceived behavioral control is the person's perception of how easy or difficult it is to perform the behavior. It is a function of control beliefs, related to the presence of factors facilitating performance (like resources and abilities) and the absence of factors hindering it (like obstacles).

There is no standard questionnaire for TPB studies [41] because each study can focus on entirely different behaviors. Researchers have applied it to varied activities such as time in screen versus time in physical activities [58], use of transportation alternatives [59], and alcohol consumption [60], to name a few examples. However, there are manuals with instructions for creating TPB-based questionnaires, such as [61]. We used the cited manual to construct a questionnaire to assess all TPB variables focused on the behavior of estimators defending (action) estimates (target) of software tasks or projects (context) when facing unreasonable pressure to change them or to accept unrealistic commitments (time).

Now that we understand more about the negotiation methods and a theory that can help us with understanding behavior change, we examine the intersection of negotiation and estimates in Software Engineering in the next section.

2.8. Negotiation and Estimates in Software Engineering

In the preliminary reviews we conducted, we found related work about the problem of changes of estimates ([7], [62]) and about the use of negotiation in the context of estimation

([17], [63]). We explore these studies in this section. One study about deliberate changes of software estimates is by Magazinius, Börjensson, and Feldt [7], which investigated distortions of software estimates, dividing them in unintentional and intentional ones. Cognitive biases, like the anchoring effect [64]—people's tendency to be influenced by the information presented before the estimation process—are examples of unintentional distortions. However, they claim that it is necessary to investigate other software estimation distortions to improve estimation accuracy: intentional ones. These distortions are alterations to the estimates made to fulfill objectives beyond the estimation context, and therefore are what we call in our text of changes to estimates. We preferred to use "changes of estimates" instead of "distortions of estimates" as the last expression carries a more negative meaning and can imply the estimator is willingly to make the change (thus requiring the use of the modifiers "intentional" or "unintentional" to clarify). The expression "changes of estimates" is neutral and requires less explanation to understand.

Magazinius, Börjensson, and Feldt [7] do not propose an intervention. They investigate intentional distortions of estimates more deeply, finding out that they may come from the developers' side [7]. For instance, developers can increase estimates for functionality that they regard as unnecessary to make project planners more likely to remove or postpone its implementation. Intentional distortions may also happen when upper management challenges the estimates given by project managers or developers, pressing them to reduce their estimates. They want the development to be cheaper, and they have better argumentation than developers. Another cause for intentional distortions in estimates is what they call of negotiation—a situation where involved people discuss estimates, and one side decreases their estimates while the other increases them [7]. In another study, the authors showed that distortional behaviors related to software project estimates are frequent [62]. Their work acknowledges the existence, importance, and impact of intentional estimate distortions. However, they lack proposals to deal with them.

In our preliminary review, we also found two studies related to the incorporation of negotiation into the estimation process. In the first one, McConnell [17] proposed the use of principled negotiation [18] as a negotiation technique developers can use to defend unpopular schedules and other estimates. The author discusses how principled negotiation can be used in the estimation context, primarily when executives and higher management challenge estimates, providing tips on the use of the method. However, the author does not provide empirical evidence about its successful application and whether it helps avoid changes in estimates—in other words, there is no evidence that its proposal changed behavior of practitioners.

In the second study, Ochoa, Pino, and Poblete [63] proposed CEBON (Collaborative Estimation Based on Negotiation), a method for estimating software cost and duration. They claimed to include negotiation as a vital part of the process. The authors evaluated the method by comparing the estimates that students generated with the ones from the courses' instructors using Wideband Delphi. The authors considered the results quite accurate. However, CEBON focuses on helping the software project team reach internal consensus through the method and the tool. It does not provide guidance about how to deal with external pressures that can lead the team to change their estimates. Another issue is that there is no discussion about the theoretical foundations of negotiation theories of the technique. Therefore, it misses the benefits of considering what this multidisciplinary topic has to offer.

Table 2.1 summarizes the abovementioned studies. We also indicate their intervention focus: whether they propose an intervention and what it is about. Additionally, we indicate whether the existing interventions are based on negotiation theories and whether they were empirically evaluated.

Table 2.1 Related Work Overview

Study	Intervention focus	Intervention based on negotiation theories?	Empirically evaluates the intervention?
Magazinius, Börjensson and Feldt [7]	No	No	No
McConnell [17]	Defending software estimates	Yes	No
Ochoa, Pino and Poblete [63]	Estimation technique, which includes negotiation	No	Yes

Considering these three studies, we envisioned the research opportunity to address changes to estimates due to pressure and the lack of knowledge and abilities on the part of software professionals on how to defend their estimates. We also realized that no one was genuinely adapting the existing negotiation methods to the estimation context to allow software practitioners to gain easier access to such knowledge. Therefore, our research project differentiates from the previous ones by focusing on:

- How software engineers can use negotiation theories to provide them with the knowledge that can help to defend their estimates and negotiate commitments with other stakeholders; and
- Gathering empirical evidence about practitioners' perceived usefulness of the defensive strategy and their intentions to adopt it in the real world.

2.9. Summary

This chapter presented the background, defining an estimate and other related concepts, such as targets, commitments, and plans. We discussed relevant concepts and how estimates and targets sometimes collide, leading project stakeholders to pressure for changes in estimates and to unrealistic commitments. We also discussed the definition of negotiation and what makes a situation have characteristics that can benefit when handled as a negotiation by everyone involved. We presented the two dominants negotiation models: distributive and integrative. Then, we summarized the three negotiation methods that form our theoretical foundation, all of them based on the integrative model. Such methods lead negotiators to focus not only on the substance of negotiation but also on preserving and improving the relationship among the parties. This characteristic makes them perfect for the context of estimation, where estimators typically deal with their bosses and clients, and wish the maintenance of a good reputation. Moreover, estimators and other project' stakeholders generally share a few interests, such as increasing organizational profits, while also keeping other individual, and possibly conflicting, ones. For instance, estimators might want to keep an estimate to avoid overtime work, while other stakeholders might be interested in delivering a product on a specific deadline regardless of how much overtime work is needed. The methods we presented were created to help people in this very specific situation of some shared and some individual interests. Therefore, they seem suitable for the estimation context.

We also explored a bit of the Theory of Planned Behavior, considering it can help us to understand more about behavior change, given that our proposed approach ultimately aims for that. Finally, we also discussed the related work, showing the existing gap for empirically evaluated approaches to improve the abilities of software estimators in defending their estimates and negotiating realistic commitments.

Moreover, we focused on understanding more of the research problem from the perspective of the existing software engineering research literature, before proposing an artifact. Instead of focusing on the factors leading to deliberate changes of software estimates, we amplified our target to factors affecting software estimates in general. We expected this would lead us to studies about changes of estimates and give us the benefit of identifying other related factors that we were unaware of at the start of this research project, like establishing and fulfilling commitments. Therefore, we executed an SLM about factors affecting software estimates, which we detail in the next chapter. In the context of the SLM, we decided to narrow our focus to expert judgment to reduce scope and because it is the preferred method in the software industry [31], [32], [33]. Expert judgment is also on the rise as a research topic in software effort estimation [34].

CHAPTER 3 – SYSTEMATIC LITERATURE MAPPING ABOUT FACTORS AFFECTING ESTIMATES

This chapter presents a systematic literature mapping to identify the existing evidence regarding the factors affecting estimates, as part of our DSR relevance cycle. Its results also are a contribution to the knowledge base (DSR rigor cycle), aiding with our problem investigation from the perspective of the SE literature.

3.1. Introduction

Expert judgment differs from other estimation methods because the quantification step for generating the estimate is judgmental rather than mechanical [65]. That is, experts use their human mind as a measurement instrument [66]. The processes used for arriving at a prediction are largely unconscious [67]. Therefore, discovering and understanding the factors that affect expert judgment estimates is crucial for reducing errors and improving accuracy when using such a method, and research on these factors is also a trend [34]. In addition, research and practice in 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 [68]. 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 [69]. 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 chapter, we provide such perspective of factors through a Systematic Literature Mapping (SLM) using the guidelines of Kitchenham, Budgen, and Brereton [70] and Petersen et al. [71].

We found 131 relevant papers in our SLM. From these papers, we extracted 235 different factors—a myriad of diverse elements that somehow influence estimation results using expert judgment. Most of these factors (166 out of 235) were reported in a single paper and as such we decided not to include in this chapter. We provide them as part of our supplementary material [72]. Still, understanding the remaining 69 factors which were reported in two or more papers 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 [73].

Likewise, the SEXTAMT uses reference points in the form of dimensions, which the interested reader can use to navigate these wide seas of factors: temporal, stakeholder, and type of effect dimensions. A temporal dimension alludes to the importance of time for calculating correct positions when using the physical sextant. In the SEXTAMT, the temporal dimension 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 [74]. Most of the factors we found group up 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, 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.

3.2. SLM Protocol

We started the SLM by defining a systematic mapping protocol, following the guidelines presented by Kitchenham, Budgen, and Brereton [70], and Petersen et al. . Next, all researchers inspected the protocol in search of improvements. The remaining of this section presents the SLM research questions. It also presents the search, selection, extraction, and analysis procedures.

3.2.1. Research Questions

The primary research question in this SLM is: ***RQ 1 – What is the existing evidence about 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 (SQs):

- **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?**[4]
- **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?**

This chapter answers SQ 1.1 and SQ 1.2 in Sections 3.3.1 and 3.3.2, respectively. We also present an organization of the factors in the form of the SEXTAMT in Section 3.4. We present the answers to all the remaining questions in APPENDIX A.

[4] These variables can be software size, effort, cost, duration, or productivity. This question can help identifying the variables that researchers concentrate their investigation of factors, contributing with a better comprehension of primary studies' scope.

3.2.2. Search and Selection

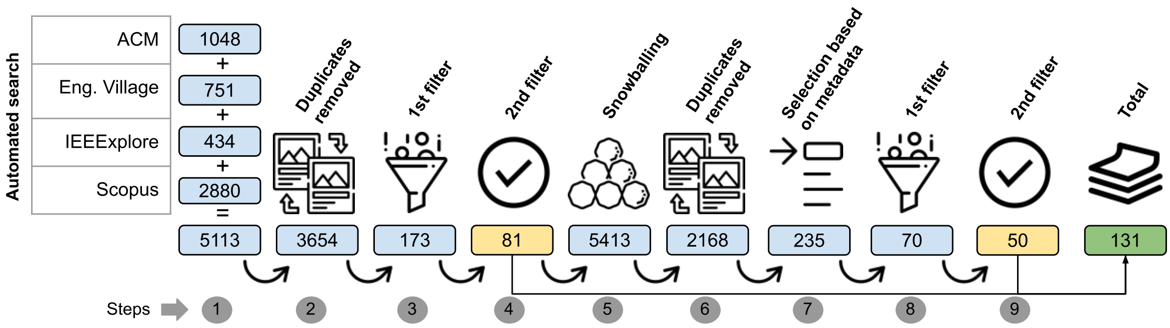
We started the search process by defining a known set of papers, which we used as an oracle (known set of papers) to validate our search string's outcomes. Our oracle had 25 papers[5]. Our next step was defining the search string. The results of automated searches are highly dependent on the search string's quality [71], [75]. We defined ours based on the extraction of the keywords of the titles and abstracts from the papers in our known set of papers, as Petersen et al. [71] recommend.

We executed the automated search restricting the search to title, abstract, and keywords whenever possible. Our sensitivity[6] goal for the automated search was 70%, as Zhang et al. [75] 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[7]. We refined the search string, leading us to the second and final version, presented in Table 3.1.

Table 3.1 - 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)

We carried out the automated search on ACM, IEEExplore, Scopus, and El Compendex (Engineering Village), as illustrated in Figure 3.1 (Step 1), resulting in 5,113 papers 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. [76].



[5] The final list with the known set of papers is in the supplementary material, together with more details of the search and selection procedures [72].
[6] Number of relevant studies (which is the number of papers in the known set of papers) retrieved in the search divided by the total number of relevant studies (again: the number of papers in the known set) and then multiplied by 100 [75].
[7] For ACM alone we had over 480,000 results.

Figure 3.1 - Search and selection results

After eliminating duplicates from the 5,113 papers, we came to a total of 3,654 papers (Figure 3.1, Step 2). Next, we executed the selection procedures, applying two filters. First, we selected papers based on their title and abstracts. Next, we read the full text. We considered 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 Table 3.2 shows. Additionally, Table 3.2 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 3.2 - Exclusion criteria and 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 and authors did not answer to contact attempts.	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, two researchers independently analyzed a random sample of 20 papers retrieved by the search by reading their titles and abstracts. We calculated the researchers' level of inter-rater agreement on this sample of papers through the kappa coefficient [70]. We got a kappa level of 0.83, which is very good, according to Kitchenham et al. [70]. Therefore, we proceeded and selected 173 papers based on title and abstract (Figure 3.1, Step 3). After reading the full text of all the 173 papers, we selected 81 that satisfied the inclusion criteria and that we could not eliminate with our exclusion criteria (Figure 3.1, Step 4).

The final set of papers selected from the database search formed the start-set for backward and forward snowballing [77]. We aimed for a sensitivity of 100% after the snowballing step. We got to a total of 5,413 papers through backward and forward snowballing (Figure 3.1, Step 5), and to 2,618 after removing duplicates (Figure 3.1, Step 6). We executed backward snowballing manually and forward snowballing with the aid of the tool Publish or Perish[8]. We selected a total of 234 of them based on their metadata—title, authors, and venue—and on their citation context on the original papers in the case of backward snowballing (Figure 3.1, Step 7). We filtered them based on their abstracts, reducing the number to a total of 70 papers (Figure 3.1, Step 8). Following, we read their full text, leading to the inclusion of 50 papers (Figure 3.1, Step 9). Therefore, the final list of papers included in our SLM contains 131 papers, and we satisfied our goal of 100% sensitivity regarding the known set of papers.

3.2.3. Data Extraction

We extracted the data using a form[9] 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 classification of Storey et al. [78], 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; and
- how authors measured the impact of the factors over the estimates.

3.2.4. Data Analysis and Synthesis

Figure 3.2 provides an overview of our data analysis. After reading the full text of all selected papers and extracting text and data to our extraction form, we created codes to

[8] https://harzing.com/resources/publish-or-perish
[9] The form, as well as the complete extraction data are in the supplementary material [72].

summarize the findings from the primary studies[10], supporting the aggregation of data into factors later during the analysis process.

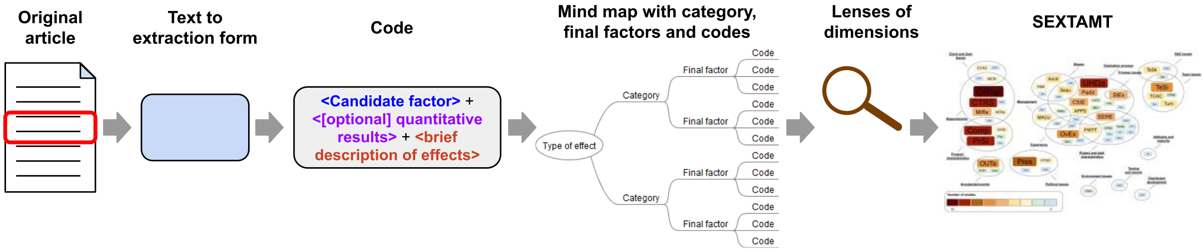


Figure 3.2 - Overview of the analysis

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 among the researchers involved, 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, each composed of categories. The categories we used to organize the data relate to three dimensions, shown in Figure 3.3.

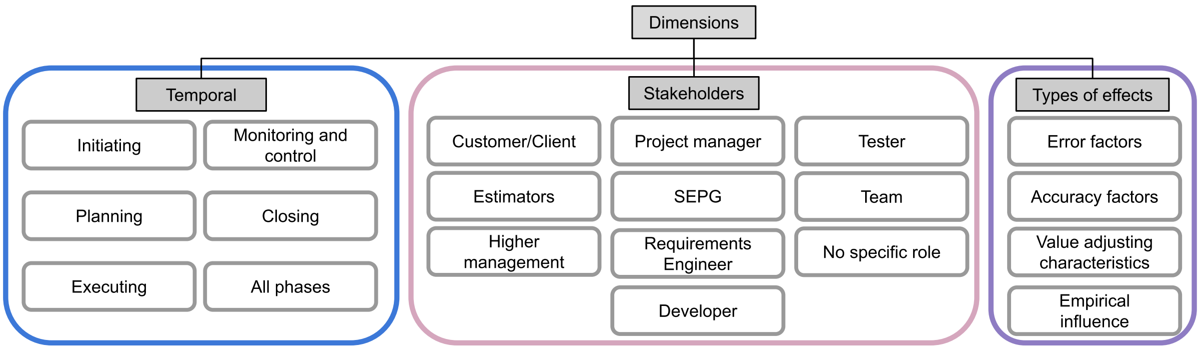


Figure 3.3 - SEXTAMT dimensions

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 [74]. The stakeholder dimension informs one stakeholder or a group responsible for a task or process 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

[10] All factors with their categories and codes are in the supplementary material [72].

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 estimates is 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 luck, such expectations are realistic, their impact are on the direction of making the estimate more accurate. Otherwise, they 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 paper to reduce space, reporting them in our supplementary material. In the next section, we explore our results.

3.3. Results

3.3.1. 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 paper. Table 3.3 presents the 69 factors, with an ID code in parenthesis, and the papers with the evidence about them.

Table 3.3 - List of factors and papers.

Factor	Papers
Anchoring effect (Anch)	[79] [80] [64] [81] [82]
Anticipation of project' participants' skills (APPS)	[83] [84] [85] [86] [87]
Availability of knowledgeable/competent clients (AKCC)	[88] [89]
Business area (BuAr)	[90] [91]
Clarity of client's needs (ClCN)	[85] [88]
Client's expectations (ClEx)	[92] [93]
Clear requirements specification (CRSp)	[83] [85] [94] [33] [95] [96] [89] [97] [98] [99]
Changes to requirements or scope (CTRS)	[83] [33] [86] [100] [89] [101] [99] [102] [87] [85] [88] [96] [89]
Collaboration and communication (CCAC)	[85] [103] [87] [104]
Combination strategy of individual estimates (CSIE)	[105] [106] [107] [108] [109] [110]
Complexity (Comp)	[94] [111] [112] [113] [114] [115] [101] [116] [117]
Cultural diferences (CuDi)	[118] [117]
Dependencies between user stories/backlog items (DUBI)	[94] [117]

Factor	Papers
Diligence (Dili)	[119] [85]
Effect of more and/or irrelevant information (EMII)	[92] [86] [120] [121]
Enough effort and resources spent on estimation (EERE)	[83] [122] [85] [96] [82] [123]
Estimation experience (EsEx)	[123] [117]
Estimation skills (EsSk)	[124] [102]
Experience with similar/previous projects/tasks (ExSP)	[96] [103]
Familiar problem or requirements (FPRe)	[100] [82]
Familiarity with the product (FWTP)	[114] [125]
Familiarity with the technology (FWTT)	[119] [114] [95] [82] [102]
Goals and targets (GATa)	[124] [7]
Impact of early estimates (IEEs)	[122] [126]
Incorrect assumptions (InAs)	[94] [95] [82]
Informal basis for estimating (IBEs)	[33] [127] [102]
Integration and dependencies (InAD)	[118] [112] [33] [114]
Involvement of technical staff (ITSt)	[83] [85] [128]
Longer projects (LoPr)	[129] [90]
Manager experience (MgEx)	[113] [97]
Misunderstanding of requirements (MiRe)	[94] [7] [88] [96] [82] [102]
Monitoring and control (MACo)	[83] [113] [96] [89] [102]
Negotiations games in estimates (NGIE)	[7] [130]
New team members (NTMe)	[83] [94] [102]
Non-functional requirements (NFRe)	[33] [114] [115] [103]
Occurrence of unforeseen problems (OUPr)	[83] [94] [82]
Optimism (Opti)	[131] [7]
Overall experience (OvEx)	[118] [33] [113] [7] [88] [96] [132]
Overlooked and unplanned tasks (OUTa)	[85] [94] [95] [7] [96] [82] [87]
Padding (Padd)	[124] [85] [96] [87] [130] [133]
Platform (Plat)	[129] [91] [117]
Pressure (Press)	[83] [85] [7] [98] [102] [130]
Price-to-win issues (PTWI)	[83] [33] [7] [96]
Programming language (Prog)	[90] [91]
Project flexibility (PrFl)	[96] [89]
Reestimation and revision of estimates (REEs)	[86] [129] [100]
Resources dependencies (ReDe)	[94] [86] [100]
Risk assessment (RiAs)	[83] [113]
Sequence effects (Sequ)	[134] [135] [136] [137]
Size (PrSi)	[94] [33] [86] [129] [100] [115] [90] [138] [103]
Similarity with previous tasks/projects (SWPP)	[96] [82]
Simplicity (Simp)	[96] [82]
Standards in estimation (StEs)	[83] [124] [85] [123] [128] [87]

Factor	Papers
Task size (TaSi)	[111] [139]
Team Collaboration and communication (TCAC)	[83] [118] [86] [88] [117]
Team Size (TeSi)	[94] [129] [115] [90] [91] [117] [139]
Team Skill (Skil)	[118] [33] [103]
Team Stability (Stab)	[33] [115]
Technical experience (TeEx)	[94] [140] [117]
Technical skill (TeSk)	[131] [95] [96] [141] [102]
Time frame size (TFSi)	[142] [143]
Tool support and avaliability (TSAv)	[114] [96]
Type of project (TyPr)	[90] [117]
Training in Estimation (TrEs)	[83] [123]
Turnover (Turn)	[85] [144] [112] [33] [145]
Unit effects (UnEf)	[135] [146]
Use of checklists (UsCh)	[111] [95] [96]
Use of flexible/agile development model (UFAM)	[147] [148] [149]
Use of historical data (UHDa)	[83] [124] [85] [94] [114] [95] [96] [123] [150] [87]

Section 3.4 details the factors, presenting them as part of the SEXTAMT. We also organized the factors considering the dimensions we presented in Figure 3.3.

3.3.2. 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. We extracted the measurement strategies as reported in the original papers by the researchers. Table 3.4 presents the associations between the strategy that researchers used for impact measurement with each paper. Each paper could have multiple different ways to measure impact.

Table 3.4 - Measurement strategies and papers.

Factor	Papers
Difference of estimates	[134] [122] [151] [79] [80] [92] [131] [152] [153] [154] [155] [156] [129] [126] [157] [64] [158] [159] [81] [160] [142] [93] [120] [121] [161] [135] [149] [116] [91] [162] [150] [163] [141] [136] [146] [164] [143] [165] [166] [139] [109] [137] [142] [110]
Participants' perception	[83] [84] [118] [122] [151] [124] [167] [85] [94] [144] [111] [112] [33] [168] [86] [169] [170] [114] [126] [7] [171] [115] [172] [88] [96] [89] [82] [123] [173] [174] [97] [103]

Factor	Papers
	[128] [101] [117] [62] [175] [98] [99] [176] [145] [127] [87] [130] [133]
MRE (Magnitude of Relative Error)	[147] [119] [131] [177] [171] [96] [178] [82] [132] [107] [179] [108] [180] [181]
MREBias (Magnitude of Relative Error – Bias)	[147] [96] [82] [182] [108]
BRE (Balanced Relative Error)	[147] [86] [106] [183] [104] [110]
BREBias (Balanced Relative Error – Bias)	[147] [119] [111] [86] [95] [106] [183] [110]
Deviation	[184] [122] [185] [186] [140] [145] [187]
Absolute error	[151] [107] [141]
Total effort	[171] [90] [138] [188] [189] [125]
Interval of over/underrun (over/underestimation)	[118] [85] [190]
Pred(X)	[119] [191]
Confidence related	[167] [105] [64] [192] [121] [82] [135] [193] [179] [194]
Not informed/not defined	[195] [196] [197] [113] [148] [102] [198] [199] [200]
Other	[151] [201] [100] [202] [203] [178] [204]

Researchers' most used strategy for investigating the impact of factors was participants' perceptions: 45 papers 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 APPENDIX A. 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 Table 3.5 shows.

Table 3.5 - Objective metrics of accuracy, bias, and error.

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

While the critiques of MRE (|actual - estimated| / actual) and MREBias ((actual – estimated) / actual) 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 in the denominator—seven studies use the estimated value at the denominator. We categorized these studies under the term “deviation”, since the researchers of such papers disagree about the best name for the metric, calling it effort deviation [184], [187], effort overrun [122], accuracy [185], effort variance [186], overrun factor [140][11], or project overrun [145]. Another three studies use the absolute error (estimated - actual value).

A total of six studies evaluates total effort. They are either based on regression analysis ([90], [171]) or correlations of effort with other variables ([125], [138], [188], [189, p.]). 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 ([85], [118]). The other study was a data one ([190]). Also, two studies used pred(x) [205].

3.4. The SEXTAMT

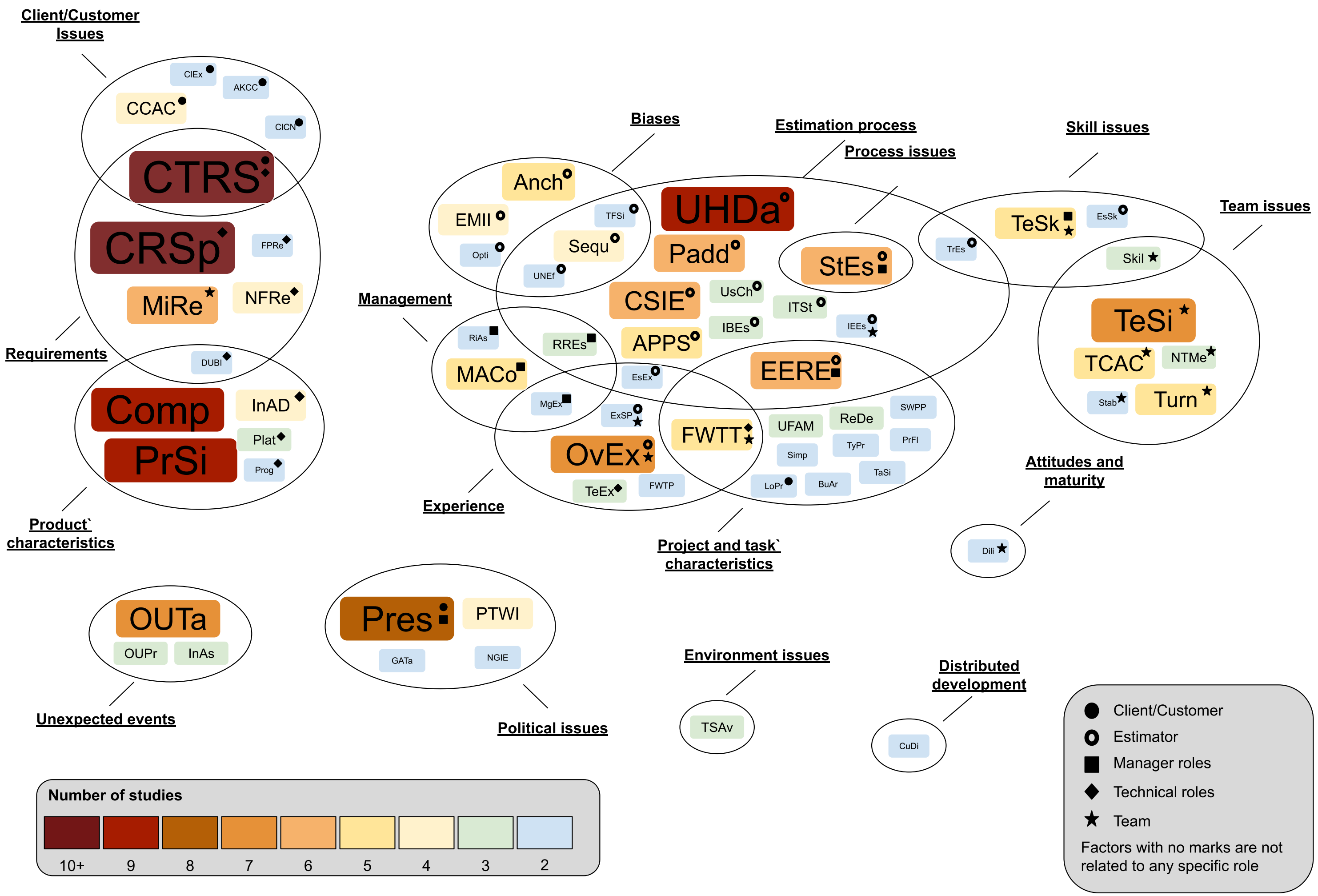
As we informed in Section 3.1, we found a total of 235 factors, of which 69 were reported in two or more papers. We gathered these 69 factors in one instrument: the SEXTAMT. It has three dimensions to allow the navigation through the seas of factors:

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.

Figure 3.4 presents 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 Table 3.3 (Section 3.3.1). We marked some of them with symbols related to their stakeholders’ role dimension—the temporal and type of effects dimensions were suppressed to increase readability. The size and color of each factor represent the number of papers investigating it. We also grouped them by major categories represented in the form of ellipses. In addition, we provided an expanded view of Figure 3.4, as part of our supplementary material, in which we added the studies that investigated each factor.

[11] 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.

AKCC	Availability of knowledgeable/competent clients
Anch	Anchoring effect
APPS	Antecipation of project' participants' skills
BuAr	Business area
CCAC	Collaboration and communication
CICN	Clarity of client's needs
CIEx	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 3.4 - The SEXTAMT

Figure 3.4 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 addition of a value to the original estimates before their communication when defining a commitment), 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: clear requirement specifications, changes to requirements or scope, misunderstanding of requirements, complexity, and product size. The map also has some category 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 3.3. In each of the following sections, 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' role and by project phase (temporal dimension). We also present the type of effect for each factor.

3.4.1. Customer/client

Figure 3.5 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 3.5. 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 3.5 by a red circle inside the factor box. Figure 3.5 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. In this dimension, we mapped the factors to the phases of PMBOK: **initiating**, **monitoring and control**, and **closing** [74]. 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 papers that published results regarding them. The numbering of each hexagon indicates the paper ID in the extraction forms (part of our supplementary material).

Figure 3.5 shows that we found no factors associated to the customer/client and that were also related to the **initiating** phase of PMBOK [74]. 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 [85] present a survey where the users' lack of understanding of their requirements is a reason for inaccuracy. Matos et al. [88] 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 [129] and that increasing calendar time will increase total effort [90]. Therefore, it is a value adjusting characteristic.

Eight studies declare that *pressure* impacts estimating, either as an error factor or as a value adjusting characteristic. The papers 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. [83] 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 [102] report that pressures from customers or managers result in lower estimates than would be realistically expected.

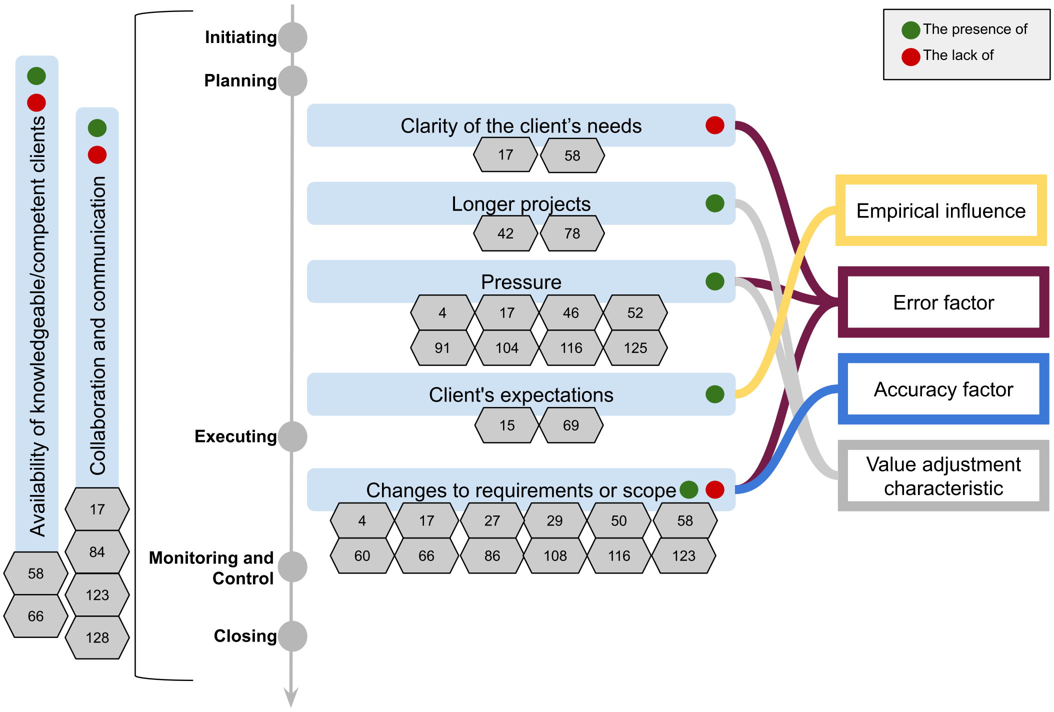


Figure 3.5 - Factors related to Customer/Client

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 [93]. This result repeated even when estimators are told to disregard such information [92].

At the **executing** phase, *changes to requirements or scope* emerge as an error factor, with twelve studies discussing it. Some studies report that requirement changes are a reason for inaccuracies [33], [100], [101], and two studies indicate that frequent changes are the problem [85], [99]. Others emphasize that requirement changes contribute to overruns [89], [140], are a challenge [86], or a potential problem for estimation [87], [102]. Finally, some researchers identify changes in scope [100] and scope creep [33], [96] as reasons for inaccuracies. When the client's needs are stable, it facilitates software estimation and raises accuracy [88], 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 [88]—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 [88], and the lack of competent customers able to make decisions is a reason contributing to overruns [89]. *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 [104]. Also, researchers found that insufficient user-analyst communication and understanding was a potential cause of estimating problems in a case study [87], confirming it is a reason for inaccuracy later on in a survey [85]. Additionally, in the agile context, customer communication is an effort predictor [103]. 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.

3.4.2. Estimators

Figure 3.6 presents all the factors for an estimator. One factor is related only to the **initiating** phase: *early estimates*—two studies indicate that they impact estimates in later phases [122], [126]. 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 [126]. 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 this, concluding that early estimates act as anchors to final estimates only when estimators have nothing to lose [122].

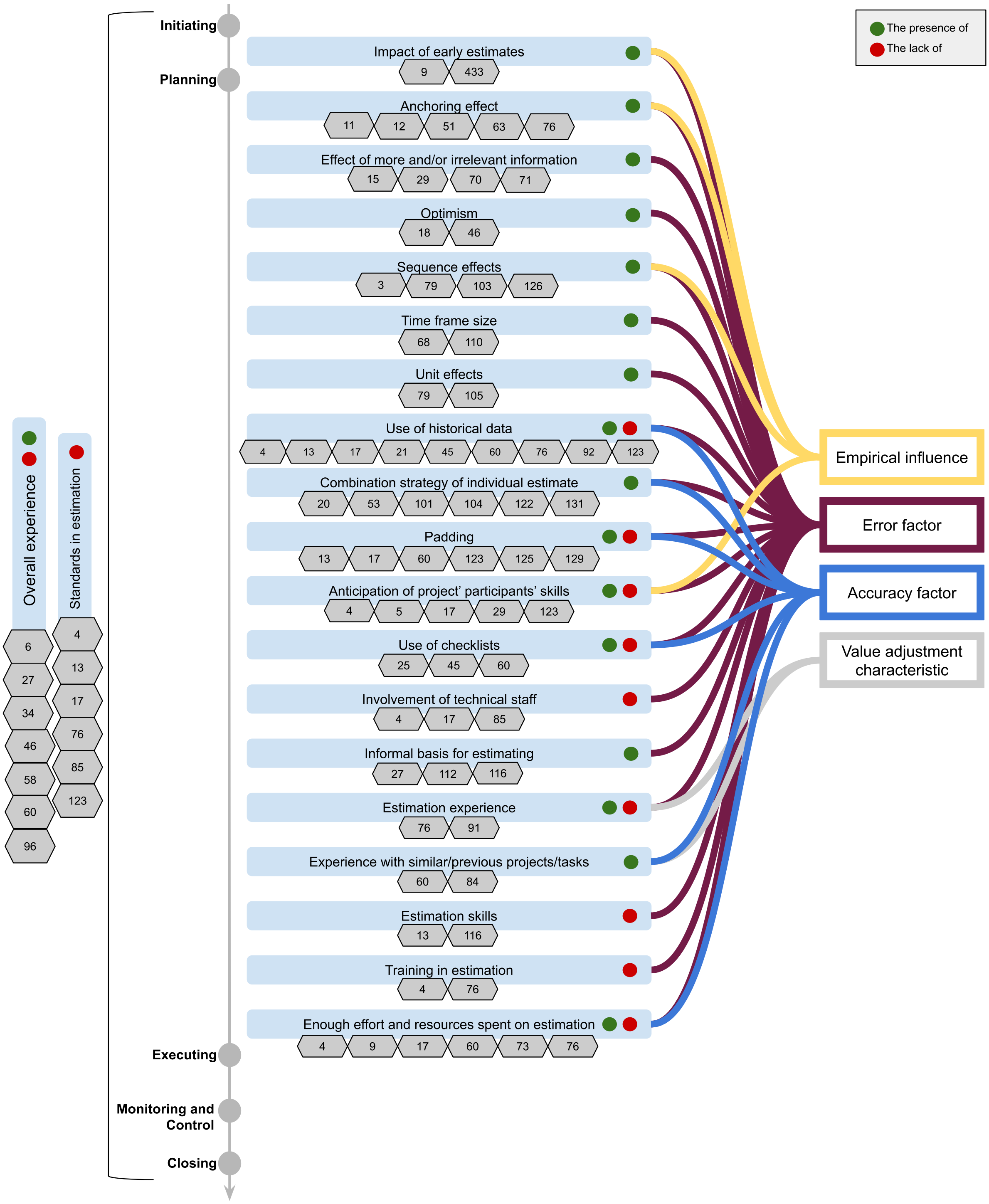


Figure 3.6 - Factors related to Estimators

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 [64]. In a field study it is reported to hinder the creation of a meaningful estimate [123] and, thus, is an error factor. Many laboratory experiments also report that the anchoring effect impacts software estimation [64], [79]–[81]—therefore providing evidence of its empirical influence over the estimates. Aranda and Easterbrook [80] found a statistically significant impact of numerical anchors on time estimates. Jørgensen and Grimstad [81] 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 [64] 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 [79] 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. [86] found that the availability of more detailed information may increase underestimation bias by increasing estimator's optimism. Grimstad and Jørgensen [121] report that specifications with irrelevant information lead to higher estimates in laboratory experiments. Jørgensen and Grimstad [92] 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 [92]. 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 [120].

Optimism is an additional error factor, leading to estimates' unintentional distortions, for instance [7]. Jørgensen, Faugli, and Gruschke [131] 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 [131].

Estimators should also be aware of *sequence effects* relative to the order of estimation of tasks and projects with different sizes[12]. Grimstad and Jørgensen [134] showed a statistically significant difference when starting estimation with a small task, compared with starting with a large one first. Jørgensen [137] 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 [137]. 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 [136].

Two papers address the *time frame size*: shorter time frames tend to lead to more optimistic estimates than larger ones [142], [143]. Another two papers investigate *unit effects*: asking for estimates using a lower granularity time unit led to lower estimates compared with using a higher granularity one [135], [146]. 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 [95]. 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 [83], [85], laboratory studies [150], and field studies [87], [94], [96], [123], [124].

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

[12] 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.

estimates, or Planning Poker. Three papers report evidence in favor of estimating in groups over averaging: unstructured group estimates [109] and Planning Poker (a structured approach) [107], [110] led to less optimistic estimates compared with the average of individual estimates. When combining interval estimates, the results also favor group discussion over averaging [105]. Mahnic and Hovelja [106] 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 estimation sessions [108]. 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 or changes in the specification is a reason for accurate estimates [96]. The greater the preference for projects completed within the estimates, the greater the padding frequency [133]. Also, the removal of padding by management is related to estimating problems [87], [124] and is a reason for inaccuracies [85]. In one study it is reported as an intentional increase in estimates, which gives it a negative denotation [130].

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 [87] and a reason for inaccuracies [83], [85]. The knowledge about who will execute testing allows for the definition of testing effort [84]. However, one study suggests that the team's knowledge of who will work on the project may increase underestimation bias [86], probably because such anticipation may not work for all contexts.

Another essential aid is the *use of checklists*, leading to a lesser magnitude of effort overruns [95]. A field study indicates that using a personalized checklist during the estimation process reduces the underestimation bias [111]. Such evidence indicates that the use of checklists is an accuracy factor. Also, the lack of checklists is a reason for estimation error [96], 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 [83], [85], [128]. Other three studies [33], [102], [127] also reported that an *informal basis for estimating* is an error factor. Lederer and Prasad [127] 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 [33], [102].

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 [117], and its absence hinders the creation of a meaningful estimate [123]. The second is *experience with similar/previous projects/tasks,* also an effort predictor [103] and a reason for accurate estimates [96]. The third factor is the lack of *estimation skills,* an estimation inhibitor [124] that can cause estimation problems [102]. The fourth is the lack of *training in estimation*, which hinders creating a meaningful estimate [123] and is a reason for inaccurate estimates [83].

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 [96]. On the other hand, making quick, rough estimates is not motivating and hinders creating a meaningful estimate [123]. Also, insufficient time, effort, or resources for estimating is a reason for inaccurate estimates [82], [83], [96], [124].

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 [132]. 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 [7], hindering software estimation and accuracy [88], being a reason for estimation error [96].

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 [87]. A follow-up survey confirms that these are reasons for inaccuracies [85]. Also, no development of estimation standards and no record-keeping of estimates and actual results make it difficult to capitalize on lessons learned [124], and no documented estimation procedure hinders the creations of a meaningful estimate [123]. Researchers also report that the lack of appropriate software cost estimation methods and processes [83] and the lack of clear guidance for estimating [128] are reasons for the inaccuracy of estimates.

3.4.3. Management Roles

Figure 3.7 presents the factors regarding management roles—including higher management, project managers, and the Software Engineering Process Group (SEPG). We explored some of them thoroughly in previous sections: longer projects (Section 3.4.1), enough effort and resources spent on estimation (Section 3.4.2), and standards in estimation (Section 3.4.2). We explore all the others in the current section.

At the **planning phase**, *pressure* came up as an error factor. Yang et al. [83] 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 [117]. Yang et al. [83] inform that the senior manager's pressure to set or change the estimate is a reason for inaccuracies—a finding that echoes in other studies [85], [98]. It leads people to change their estimates intentionally [7], to cave in to people with more power [130], resulting in lower estimates [102]. A final facet is schedule pressure, which leads to more effort in test tasks [115] —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 [113], and the lack of it is a reason for inaccurate estimates [83]. Surprisingly, some laboratory experiments' results indicate that identifying more risks immediately before software estimation leads to increased over-confidence [156]. The authors clarify that they have not investigated a complete risk management process—only the impact of simple risk identification.

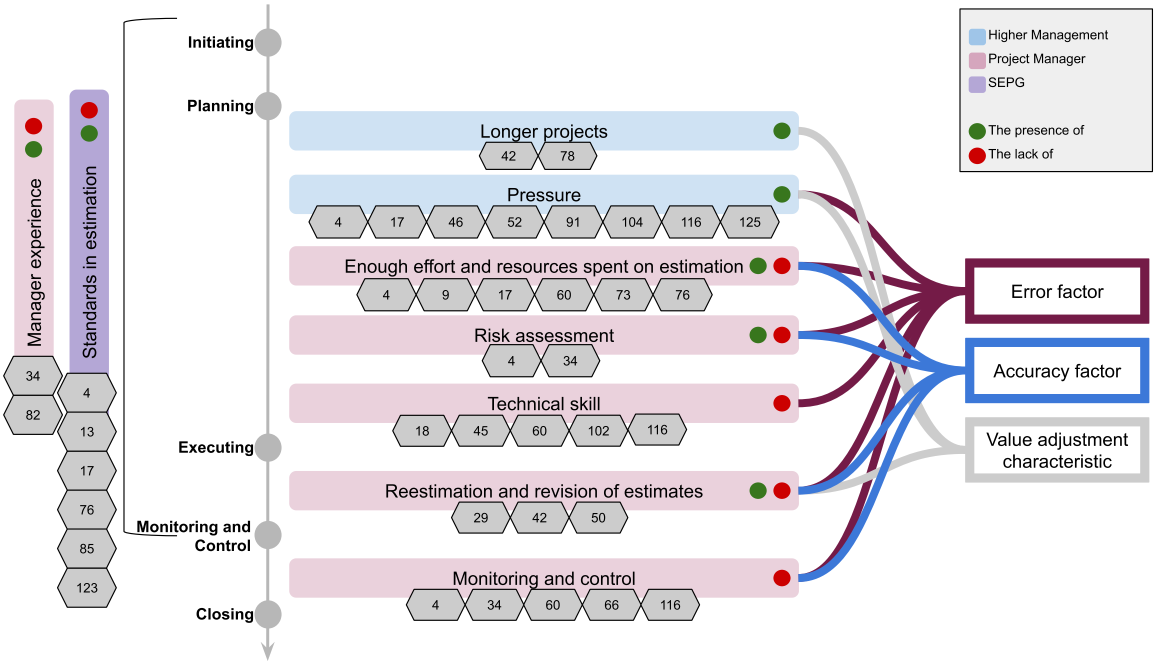


Figure 3.7 - Factors related to Managers

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 [96]. Other studies also report technical skill issues but concerning the team, and we describe them further in Section 3.4.4.

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 [86]. In a data study, more budget revisions were related to higher costs [129]—and therefore, we considered it a value adjusting characteristic. However, in another data study, more estimation updates were connected with larger errors in effort estimates [100]. 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 [96]. One a respondent study reports that adequate project administration is a reason for the prevention of overrun [89].

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 [113]. 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 [97].

3.4.4. Technical Roles

We found factors related to technical roles: requirement engineers, software designers, developers, and testers. Figure 3.8 brings such factors to the surface. None of them apply to all phases. We explained two of them in Section 3.4.1: changes to requirements or scope and pressure.

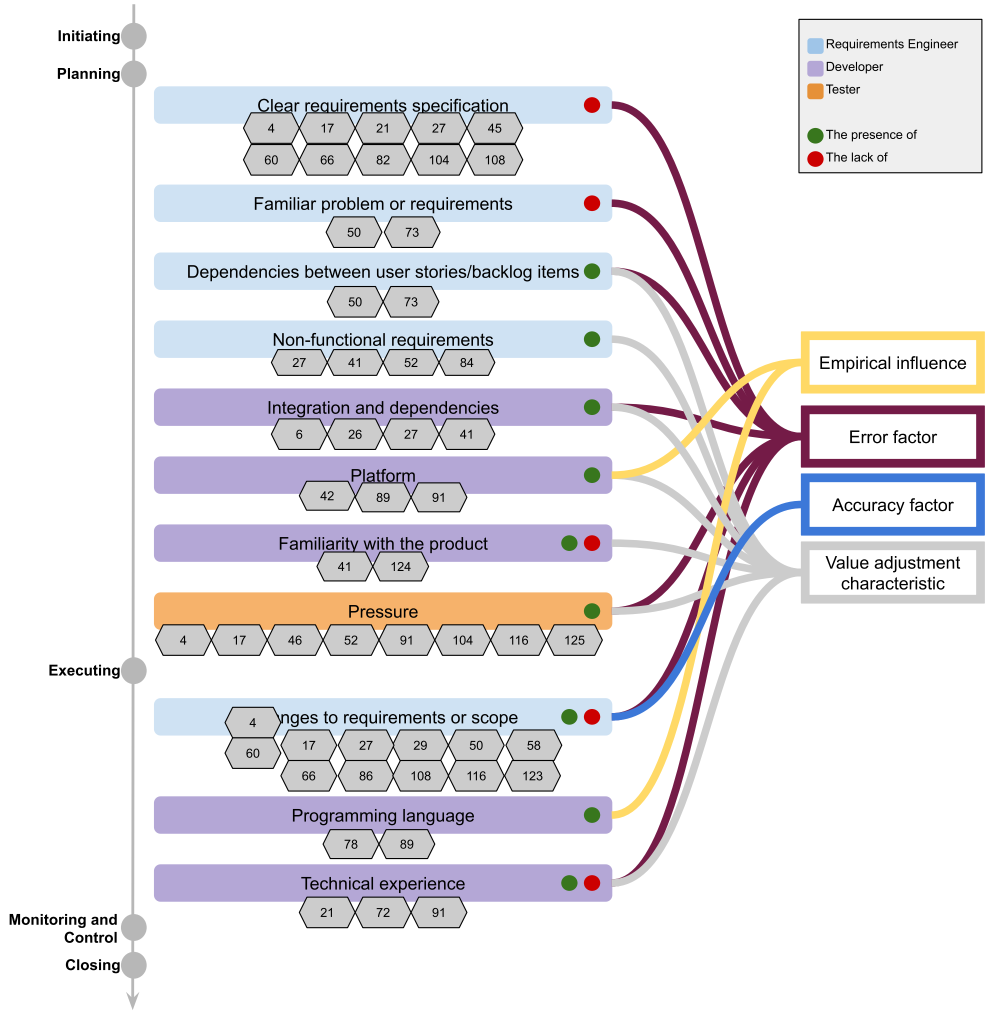


Figure 3.8 - Factors related to people in technical roles

We found four factors related to requirements at the **planning phase**, which we associated with the **requirement engineer's** 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 [33], [83], [85], [89], [96], [98]. Other studies emphasize specific facets that make requirements poor, like the redundancy of user stories [94], missing requirements [33], weak or ambiguous requirements [95], incomplete requirements [96], and the user's lack of understanding of their requirements [99]. 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. [100] report that unfamiliar feature requirements are a reason for estimation inaccuracy. Jørgensen and Gruschke [82] 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. [94] found that links to other stories serve as indicators for a possible inaccurate estimation. Altaleb, Altherwi, and Gravell [117] 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 [33], [103]. 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 [114], the required level of performance, and the required security level [115]. 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 [118]. Another one considers that integration issues are a cost driver, also in the context of agile development [33]. In the context of corrective maintenance of object-oriented systems, a high level of code/system dependencies leads to higher effort [114]. Therefore, the integration and dependencies factor is a value adjustment characteristic. Another study informs that integration complexity is an estimation challenge [112], 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 [117]. Other two studies report that the type of platform impacts software costs [129] and that the interaction of team size and development platform has a significant impact on productivity [91].

Finally, the developer's *familiarity with the product* is a value adjustment characteristic. When low, it leads to more need for effort [114]. 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) [125].

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 [90] and on time-to-market [91]. Huang, Sun, and Li [91] 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 [117] evidence that developer implementation experience is an effort predictor. Also, developers' lack of experience leads to estimation inaccuracy [94], and the lack of technology experience leads to a higher probability of effort overrun [140].

3.4.5. Team

Some of the factors we found regarded the whole software team. We show them in Figure 3.9. 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 3.4.2.

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 [114]. Other studies also indicate that the use of new or little-known technology is a reason for estimation inaccuracies [82], [95], [119] and a significant threat to estimates [102]. Also, many studies report results regarding how the *misunderstanding of requirements* leads to estimation inaccuracy and errors [82], [88], [94], [96], [102]. It also causes unintentional distortions of software estimates in different directions: either as increases or decreases of estimates [7].

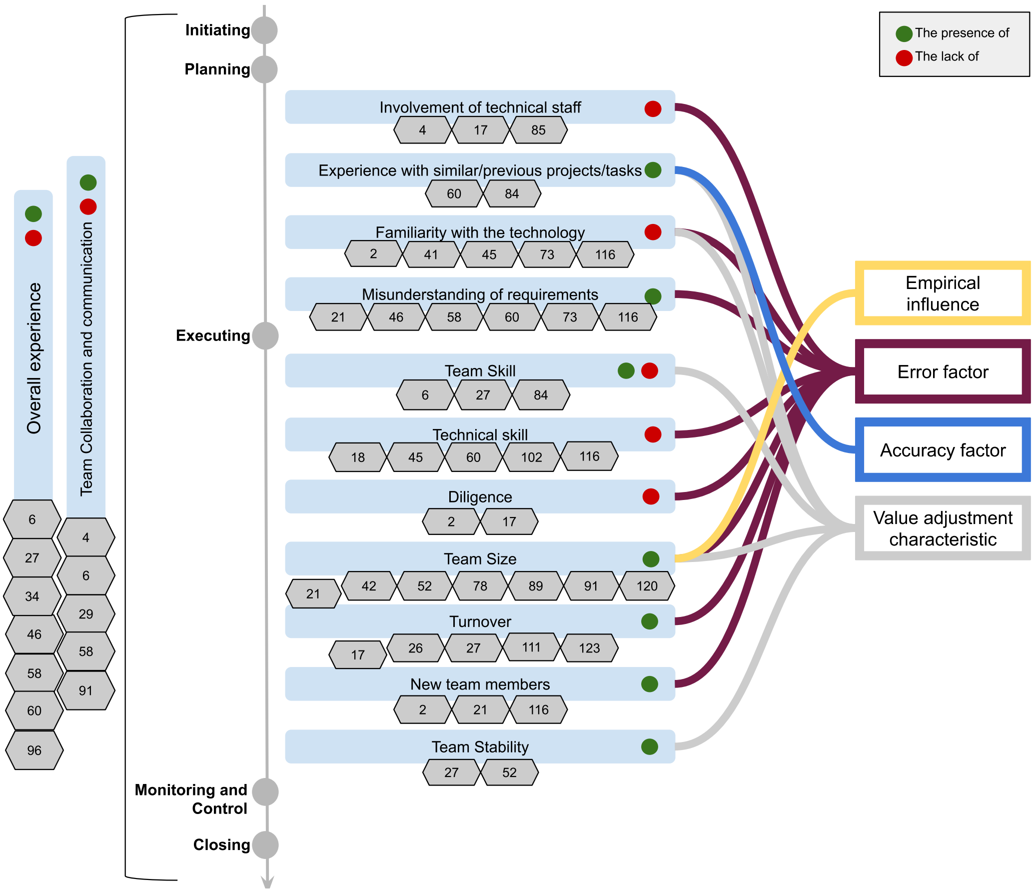


Figure 3.9 - Factors related to the team

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 [33], [103], [118]. Another more specific factor is the *technical skill*, which we partially addressed in Section 3.4.3. The presence of unskilled members in the team is a reason for inaccurate estimates [33]. Lack of technical skills [95] and technical expertise in a particular area [102] lead to estimation inaccuracies. Less software development skill is weakly connected with optimistic predictions too [131]. More specifically, Jørgensen, Bergersen, and Liestol [141] 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 [85]. 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 [119]. 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 [91], [117], and larger teams connect with higher effort and costs [90], [115], [129]. 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 [91], [117]. Interestingly, two studies suggest that multiple developers' involvement in a story or a task may lead to over or underestimations [94], [139]. 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 [33], [85], [145] and estimating problems [87]. The loss of organizational knowledge due to high turnover is an estimation challenge [112]. The existence of *new team members* leads to estimation inaccuracies [94] and a higher need for effort than estimated [119]. Another study reports that the introduction of new people is a major threat to accurate estimates [112]—and therefore, we classified it as an error factor. Finally, regarding team *stability*, one study reports it as a cost driver [33], while another one stresses that team continuity leads to less effort in the context of testing tasks [115]. 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 3.4.2. 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 [33], [118]. Another respondent study indicates that low team experience correlates with duration error, with less experience leading to more error [113].

The other factor related to all phases is *collaboration and communication*. The communication process and the communication model are effort predictors [117], [118]. On the one hand, team collaboration facilitates software estimation and accuracy [88]. On the other hand, the lack of stakeholder collaboration is a reason for inaccurate estimates [83]. Also, inherent difficulties related to communication and coordination present in multi-site arrangements lead to higher effort overruns [86].

3.4.6. No Specific Role

In Figure 3.10, 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. During the **planning phase**, *price-to-win issues* play a role when present. Price-to-win is described as an estimate defined by the price or schedule needed to win a job [206]. An estimate strongly impacted by price-to-win is a reason for estimation error [96]. Allowing the project bidding requirements to predefine the project cost [83] or purposefully underestimating the effort to obtain a contract [33] are reasons for inaccurate estimates. Magazinius, Börjesson, and Feldt [7] 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 [124], and that personal or organizational agendas lead to intentional distortions of software estimates [7].

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 [82]. On the other hand, projects frequently different from earlier projects are a reason for estimation error [96]. The *task size* is also an error factor: larger tasks are more prone to effort overruns [111], and tasks with more subtasks were underestimated compared to tasks involving fewer ones [139]. 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 [117]. He et al. [90] 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 [82], [96].

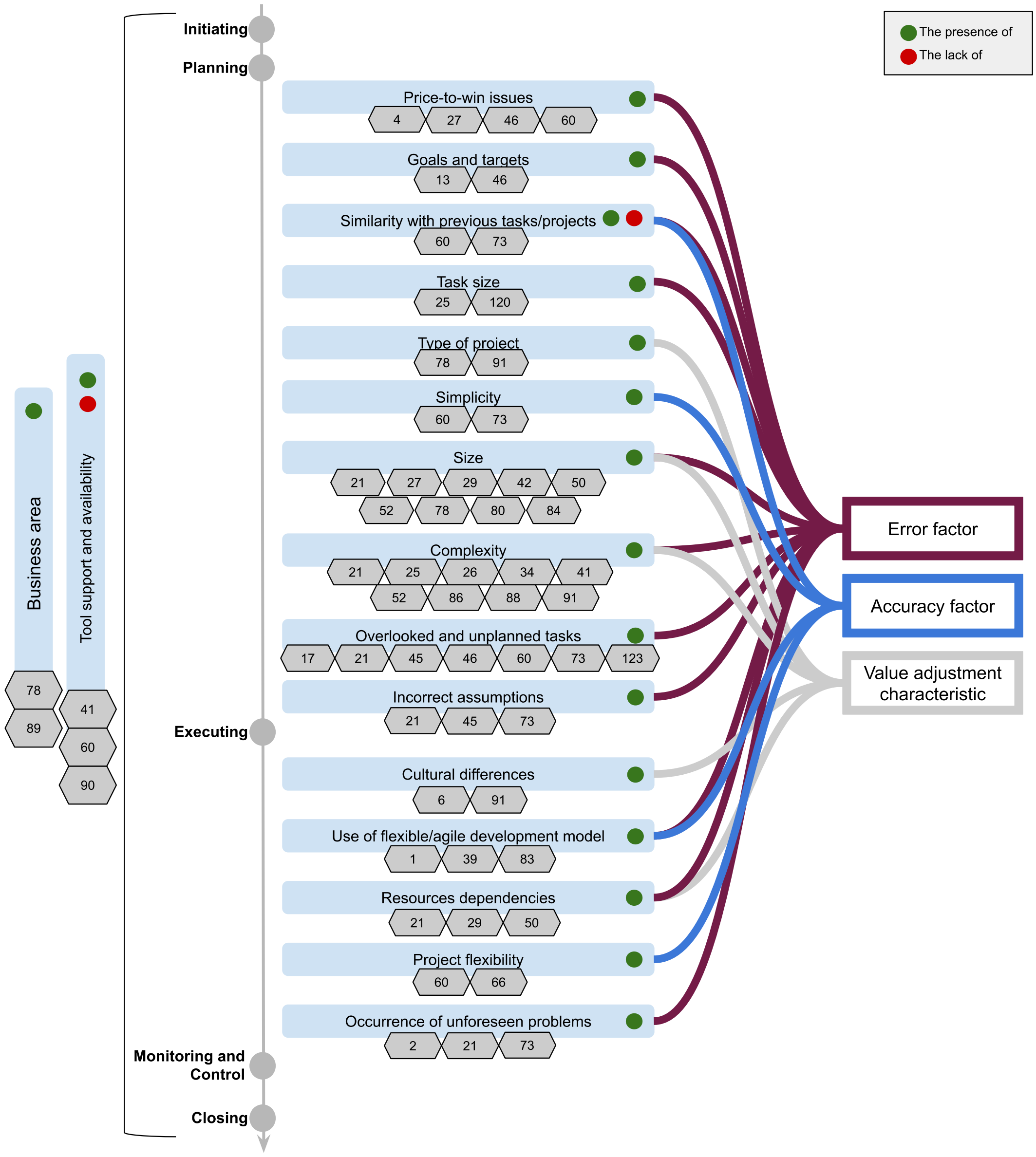


Figure 3.10 - Factors unrelated to any specific role

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 [33], [90], [103], [138]—with larger project sizes leading to more effort [129]. Size is also an error factor. For instance, Conoscenti et al. [94] 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 [100]. Finally, a field study indicates that smaller product customizations tend to be overestimated, while larger ones tend to be underestimated [86].

Complexity is a factor with many facets. Requirements complexity [115] and high technical complexity [114]–[116] leads to more effort. In the context of mobile development, one study points out that application form complexity is an effort predictor [117]. Therefore, complexity is a value adjustment characteristic. Some studies report technical complexity [101], [111]–[113] and feature complexity [94], [112] 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 [87] and a source of inaccuracies and errors [7], [82], [85], [94]–[96]. Unplanned tasks or re-work also is a reason for estimation error [96]. Closely relate4d, *incorrect assumptions* when estimating is also an error factor that may be related to the code [82], functionality [94], or complexity [82], [95].

At the **execution phase**, distributed development issues also play a role when they are present. Two studies report *cultural differences* as an effort predictor [117], [118]. 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 [147] report that flexible models are associated with lower effort overruns than sequential models. Koch and Turk [148] also report that the use of agile methods is related to less effort deviation from estimate than rigid models. However, Brown et al. [149] 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 [86], and developer resource constraints and external commitments are a reason for estimation inaccuracy [100].

Project flexibility is another relevant accuracy factor: a high degree of flexibility in implementing the requirement specification is a reason for accurate estimates [96]. 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) [89].

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 [82], [94], [119].

Two of the factors affect **all phases**. The *business area* has an impact on the effort [90] and productivity [91]. The other factor is *tool support and availability*. Software development tools have an empirical influence over management and testing efforts [162]. Additionally, insufficient tool support for project management is a reason for estimation error [96], and the low availability of required tools leads to higher effort [114].

3.5. 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.

3.5.1. The Seas of Factors that Researchers Explored the Most

The top-five factors in the SEXTAMT regarding the number of papers reporting them are *changes to requirements or scope*, *clear requirement specifications*, *product size*, *complexity*, and *use of historical data*. Most factors (40, i.e. around 58% of the total) were reported in three or more papers. The remaining 29 factors (around 42%) were investigated in two research papers only, indicating that they could benefit from further investigation.

Besides counting the number of papers, it is possible to evaluate the existing evidence for the factors in the SEXTAMT by considering the research strategies that researchers employed to investigate them. Each strategy has its inherent limitations and strengths [207]. Each one has the potential to maximize one research quality criteria at the expense of others. Respondent strategies have the potential to maximize generalizability; field strategies can maximize realism; laboratory strategies can maximize control; and data strategies can maximize precision [78]. Therefore, 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.

Figure 3.11 represents only the factors investigated in five or more papers—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).

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). These are the most explored factors in our SLM. 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*.

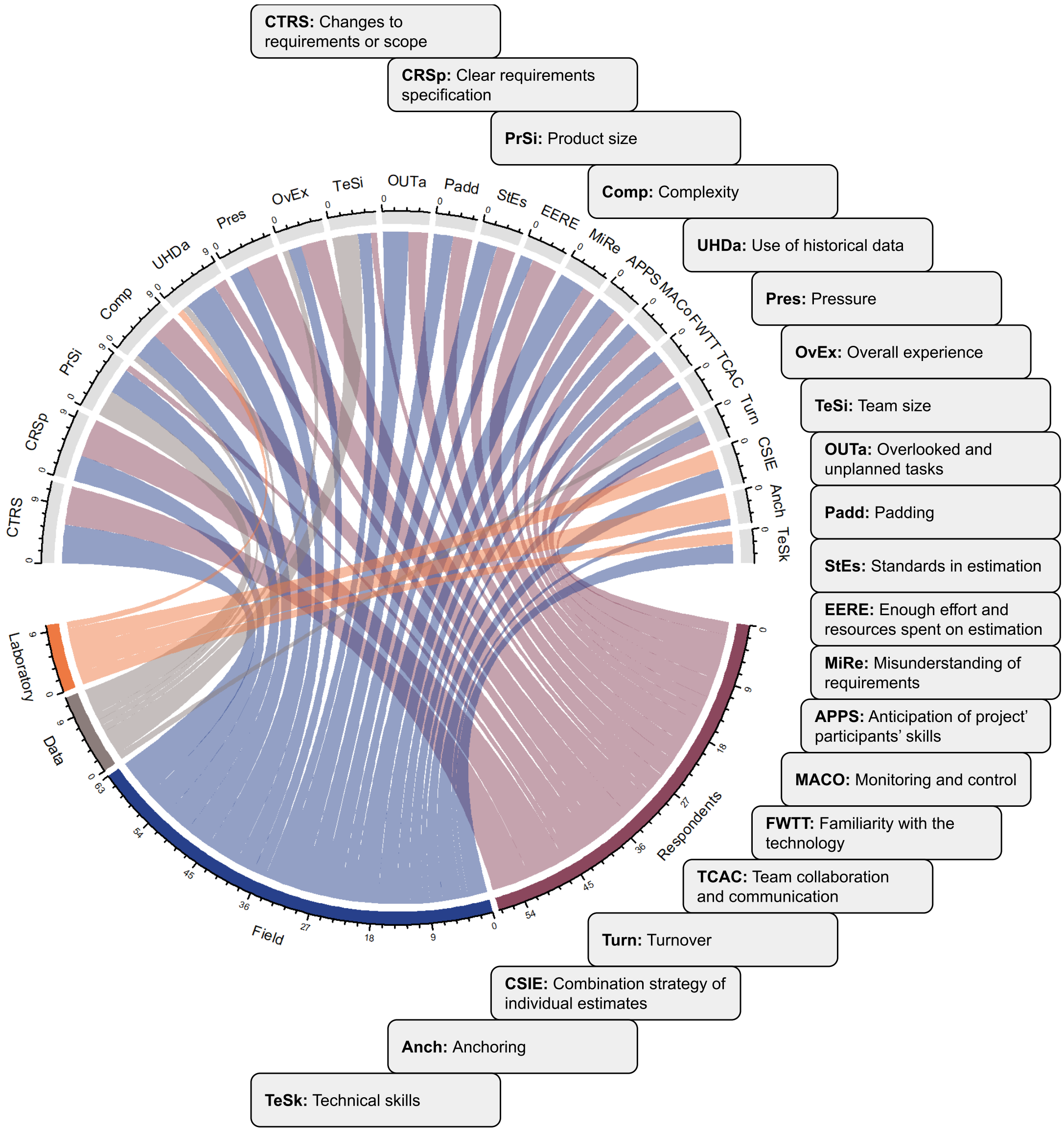


Figure 3.11 - Top factors and the studies' research strategies

All the remaining factors in Figure 3.11 were investigated using two different research strategies. Most of them involved respondents and field strategies, suggesting that the findings supporting such factors are more substantial in generalizability and realism of context. However, they may be weaker in terms of control over variables and precision of data. Some of these factors indicate that improvements in the requirements engineering process benefit greatly the estimation results, such as reducing *changes to requirements or scope* (6 R, 6 F), working on getting *clear requirements specifications* (6 R, 4 F), and minimizing *misunderstanding of requirements* (2R, 4 F). Other factors are related to improvements to the estimation process, such as *padding* (3 R, 3 F), *using standards in estimation* (3 R, 3 F), *spending enough effort and resources on estimation* (4 R, 2 F), and *anticipating project' participant skills* (2 R, 3 F). Unexpected events also have their role: *overlooked and unplanned tasks* (3 R, 4 F). Reducing such events—possibly with the *use of checklists*, another factor from SEXTAMT—is necessary.

However, improving the requirements engineering and the estimation processes might prove insufficient. Political issues such as *pressure* (5 R, 3 F) have an impact on estimation results. The management process is an additional concern: adequate *monitoring and control* (3 R, 2 F) are necessary to ensure fewer errors. Another piece in the puzzle is related to experience: *familiarity with the technology* (3 R, 2 F). Finally, team issues also have their role, through the team *communication and collaboration* factor (4 R, 1 F).

A few factors presented in Figure 3.11 involved the combination of field and laboratory strategies, suggesting their strength regarding realism and the control over variables. The first reinforces the importance of the estimation process: the *combination strategy of individual estimates* (3 F, 3 L). The existing evidence favor group estimation over averaging the individual estimates. The second factor pertains to the realm of biases affecting estimates: *anchoring* (1 F, 4 L). The third one relates to skill issues: *technical skill* (3 F, 2 L).

Many of the top factors were probably investigated extensively because of their true impact on the estimates. A few of the other factors may have been investigated because of a controversial result. For instance, despite most of the results favoring group estimation over averaging regarding the combination strategy of individual estimates, one study found the opposite when participants were students [106].

The SEXTAMT factors excluded from Figure 3.11 were reported in four or fewer papers 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. 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. [111] 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, either by surveying the literature on the search for effort or cost drivers (top-down approach) and/or the software experts involved in the estimation process (bottom-up approach). Therefore, we support their checklist build process by giving practitioners a map of factors for the top-down approach in this first step, saving them from surveying the literature themselves—a process that involves high costs.

In addition, some of the SEXTAMT factors can be helpful in the debiasing and decision hygiene strategy that Kahneman, Sibony, and Sunstein [208] proposed to help improve judgments in general. Their strategy involves decision observers: people in charge of observing others making judgments in real-time to identify whether biases are pushing them away from the best possible judgment. Decision observers use checklists to accomplish their tasks. Kahneman, Sibony, and Sunstein [208] suggest that practitioners adjust these checklists 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 in such an adapted checklist.

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 papers. 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.

3.5.2. Looking through the Lenses of 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 a single paper 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” [202]. Therefore, estimators might need to pay special attention to the initiating phase in bidding contexts.

A few papers can also shed some light on what is relevant at the **closing phase**. For instance, at least four studies ([96], [88], [131], and [82]) suggest that estimation error, feedback and learning from past projects and tasks might be beneficial to reducing overconfidence and improving estimates. Therefore, more investigation about such issues can help us to discover relevant factors at the closing phase.

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.

3.5.3. The Strategies Researchers Employed to Explore the Seas

As for the project variables, most studies focused on **effort**, which is understandable—as McConnell [44] 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. However, the approach has drawbacks also. For instance, people may attribute different meanings to the term "estimate", even when they work at the same company [204], making it difficult to interpret the results of surveys [205].

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 and errors, 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% [209], identifying such factors can be very useful. Additionally, measures such as BRE and MRE may be misleading because they depend on actual values, and work can be adjusted to fit an initial estimate [205], distorting accuracy. The same does not happen to the difference of estimates.

Additionally, researchers have used objective error measures, such as **MRE, MREBias, BRE,** and **BREBias**. 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 [205]. MREBias suffer from this same problem. BRE and BREBias are balanced metrics in this sense [147]. Figure 3.12 groups 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. Also, researchers prefer accuracy metrics over bias: with 19 occurrences for MRE and BRE together, as opposed to 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 [67].

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 [126], leading to a "moving target problem" [205] 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 [210]. 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 [211].

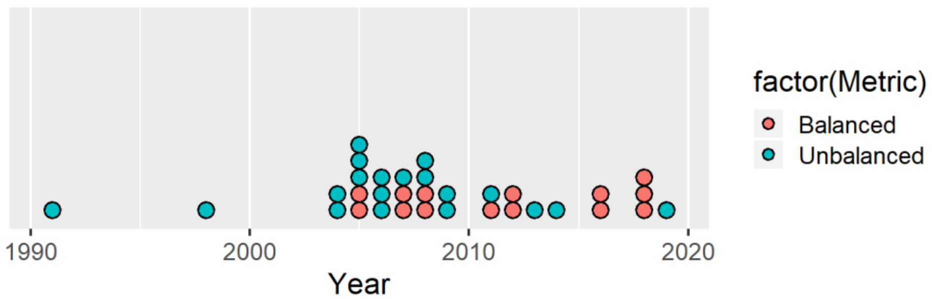


Figure 3.12 – Balanced (BRE & BREBias) x unbalanced metrics (MRE & MREBias) over the years.

Regarding our SLM results, we found very few studies discussing the variability of estimates. Only one study focuses on the existence of noise in the software estimation domain, using the term inconsistency instead. This study shows 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 [177]. In addition, very few studies in our SLM report the standard deviation of estimates (when using the difference of estimates as a measurement strategy). For instance, the study of a debiasing intervention to deal with the anchoring effect showed a reduction of the standard deviation of estimates due to the intervention [79]. In another study, researchers observed that the use of checklists and group discussions reduced the variability of size estimates, but not of effort estimates [151].

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 noise audits: experiments designed to measure how much disagreement there is among their professionals analyzing the same cases [210]—or, in our case, estimating the same tasks independently. If they realize that noise is a major problem in their context, this can be the starting point for improvement initiatives.

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 papers 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 papers 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 [81], as well as of single and interval anchors [64]. Another refinement was the investigation of the moderating effect of the expertise of the source and of the receiver of the anchor value [64] and the impact of one intervention to reduce its effects [79]. Another example is the *sequence effect*, whose impact over the estimates varies with the size of the tasks estimates in the sequence [136]. 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 [207] 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 [207] 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. However, 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 [96].

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.

3.6. Factors related to changes of estimates and to commitments

Considering that our research problem targets changes of estimates when establishing commitments, due to pressure, we screened the factors we found in our SLM to identify the ones related to these issues. Figure 3.13 presents the factors that we selected from such screening. Next, we discuss how each of these factors considering pressure, changes to estimates, the negotiation of commitments, and the lack of defense of estimates.

Customers and higher management may interpret *early estimates* as commitments, making it hard to change them later when the team carries on more detailed estimation. This does not leave much space for software teams and practitioners to negotiate more realistic commitments later on the project, when they have gained more information about their requirements.

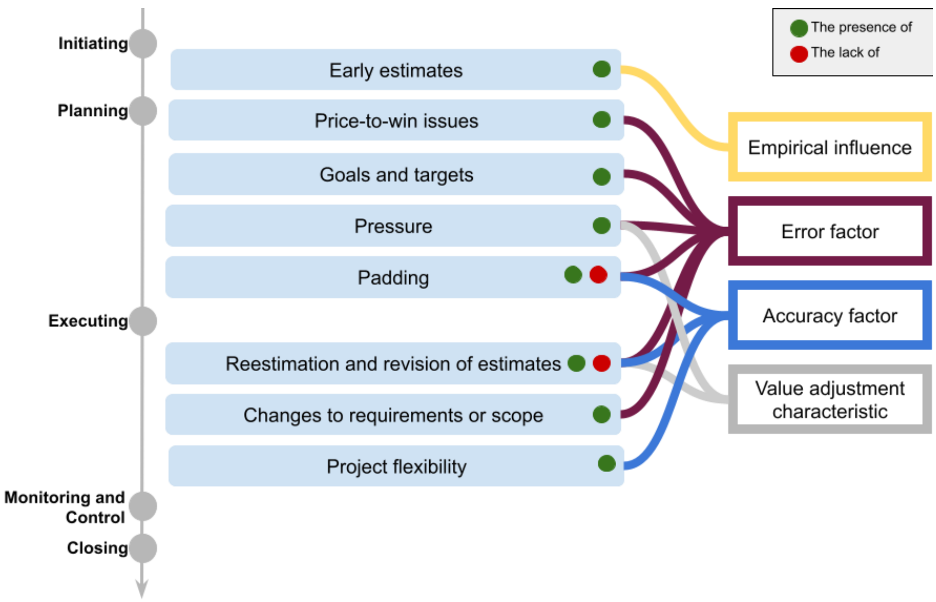


Figure 3.13 Selected Factors

Price-to-win issues involves defining the software cost estimate as the price believed to win the job [206]. It means estimates are not really predictions. *Goals and targets* involve letting personal and organizational agendas set the estimates. Although price-to-win and goals and targets does not involve a direct change to estimates, they involve presenting an estimate the estimator does not believe in given the information available, to make the commitment possible. There is not much room for negotiating commitments in any case, as well.

Pressure is related to managers or customers pressing the software team to change their estimates—a situation requiring the defense of estimates. The studies also report the removal of *padding* by management, which also requires software practitioners to defend their estimates.

Reestimation and revision of estimates may be considered an intervention to satisfy commitments, and one company can execute reestimation at a later stage to improve accuracy [86]. However, too many updates of estimates also were associated with more errors [100], probably because they may happen more frequently in projects with trouble. Another possible explanation is that they happen in the context of high pressure, with estimators facing the fact they are not going to fulfil commitments previously establish, which may lead them to adjust insufficiently during the revision. Therefore, estimators must overcome such pressure to provide new estimates that still make commitments feasible.

Some studies also report how *changes to requirements or scope* may lead to errors in estimation. This probably happens because other commitments like schedule and costs are not adjusted to reflect the changes to requirements or scope, making them unrealistic. Estimators need to defend these adjustments, therefore.

A few studies report on how *project flexibility* is an accuracy factor. It involves the possibility to reduce scope [96] or a high degree of liberty to implement the software requirements [89], which can mean delivering with lower quality. However, reducing quality or scope to meet other commitments may not be desirable in some situations, and estimators need to learn how to defend changes to commitments, instead of reducing scope or quality in such situations.

3.7. 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 [212]. 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. We needed actions to mitigate the fact that we did not test the search string in other libraries or indexing systems other than the ones we mentioned in Section 3.2.2. One of them, was the use of a known set of papers to evaluate the search strategy [75]. The goal of this evaluation was to reach a sensitivity of 70% in automated search [75]. 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 papers, 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. [78]. We used the process groups from PMBOK [74] for the phases and familiar stakeholders' roles regarding the factors. We aggregated similar findings under labels that reflected the papers' 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.

3.8. Summary

In this chapter, we presented an SLM about factors affecting software expert judgment estimates, contributing with a better understanding of our research problem. 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, once 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.

We also discussed the factors that are more closely related to our research problem: "software practitioners deliberately change their estimates because of objectives outside the estimation context, yielding to pressure over their estimates during the establishment of commitments, instead of defending them". Some of them confirm the existence of deliberate changes of software estimates and the difficulty to negotiate more realistic commitments, like the existence of *early estimates*, *price-to-win issues*, *goals and targets*, *pressure,* and *padding*. Other factors impact the satisfaction of commitments, like the *reestimation and revision of estimates*, *changes to requirements or scope* and *project flexibility*. However, only the *reestimation and revision of estimates* and *project flexibility* may be considered interventions emerging from the interaction of software estimation and the establishment of commitments. To understand more about such interaction from the perspective of practice, we conducted a qualitative study about the interaction of the software estimation process and the establishment of commitments in the software industry, which we present in CHAPTER 4.

CHAPTER 4 – THE INTERACTION OF COMMITMENTS AND SOFTWARE ESTIMATES

This chapter presents an empirical study that we execute to gain more knowledge about how estimation and the establishment of commitments interact in the software industry, as part of the DSR relevance cycle. Its results are a contribution to the knowledge base (DSR rigor cycle), aiding with our problem investigation from the perspective of the SE practice.

4.1. Introduction

Even though past research on software estimation has focused on the creation and improvement of methods [213], software practitioners still face difficulties with inadequate software estimation in the industry [214], where the most prevalent and preferred method is expert judgment [31] [32]. The literature about expert judgment is usually concerned with accuracy, primarily attempting to increase estimates' realism. As the results of our SLM show in CHAPTER 3, researchers have investigated the impact of a great variety of factors over such estimates, including human aspects such as the estimators' role [154] and level of optimism [131]. Furthermore, research findings show that expert judgment estimates may be less realistic due to judgmental biases, like anchoring [64], framing effects [192], and sequence effects [136].

Among the factors possibly affecting estimates, there is the issue of using estimates to establish commitments with management, customers, or customer representatives. External forces and pressing needs may lead software professionals to make commitments that their estimates do not support [10]. In this direction, Jørgensen [215] recommends distinguishing between the (i) PX effort estimate—that is, the estimator believes there is an X probability the value will not be exceeded; (ii) the planned effort, which is the effort used in the project plan; and (iii) and effort-to-win, which is the effort acceptable from the perspective of the market or the client. These terms help differentiate between estimation, planning, and bidding, which are processes for different purposes. Failing to make this differentiation hinders the realism of estimates. However, this distinction is not always clearly made in the industry, and when an executive asks for an estimate, what they may really want is a commitment tied to a desirable business outcome [216].

Regarding the role of customer expectations on estimates, Jørgensen and Sjøberg [93] found medium to large effect sizes in one experiment where researchers exposed participants to expected effort attributed to the projects` customers. Participants who received specifications with an exceptionally low number regarding expected effort had much lower estimates than participants who received high values. Jørgensen and Grimstad [120] also indicate that unrealistic client budgets' knowledge led to lower estimates.

The literature also reports on explicit pressures to change estimates due to customer or management expectations in industry. In the '90s, information system professionals reported that such pressure for changing estimates was associated with overruns [85]. Such results reverberate in the 2000s and 2010s, with reports about management and customer pressures leading to unrealistic estimates [102] and inaccuracies [98], intentional padding/shrinking [217], as well as estimates sometimes being a cave-in to people with more power [130].

The customer's selection strategy to choose a software provider also impacts the estimates, and there is a strong relationship between the focus on selecting developers with lower effort estimates and observed overruns [182]. In an anecdote of a real-life situation, Halkjelsvik and Jørgensen [26] comment on a software company manager who reports that their company only wins contracts when they are overoptimistic about the time to complete the work. They usually find out later that the estimate used as the basis for the price offer—their commitment—was too low. When applied to bidding scenarios, we call this the "winner's curse", which in high uncertainty situations and a high number of bidders may lead companies to have lower profit levels [202]. The winner's curse is a byproduct of the selection bias, which occurs when the client's selection process for providers leads to an over-representation of proposals based on overly optimistic estimates [182].

In this study, we further investigated the issues related to the interaction of the estimation process and the establishment of commitments in software companies, going beyond bidding situations. We explain how we executed this study to accomplish our research goal in the next sections.

4.2. Research Methodology

To investigate the interaction between the software estimation and the establishment of commitments in software companies, we performed a qualitative study, which we detail in this section.

4.2.1. Research Question

We defined the following research question for this study:

RQ - How are software estimates used to establish software development commitments?

The focus is on identifying the interactions among stakeholders to define final estimation values, including the strategies for reaching an agreement on estimates. We also look for any strategies that software professionals use during the conversion of estimates into commitments. Understanding the state of practice on this can make explicit the weaknesses on current strategies that practitioners use on negotiating commitments, thus supporting the creation of a more robust artifact for the defense of software estimates and on our path to answering this dissertation' research question (as expressed in Section 1.1).

4.2.2. Studied Organizations and Participants

This research study had two phases. In the first phase, two companies agreed to participate. We selected one of them, referred to as Company A. The main criterion for selecting the company for this phase was: they must have at least one software development team estimating their tasks regularly. We did not include the other company because their teams did not perform explicit estimation activities as part of their software process.

Company A is medium-sized, with over 100 employees, located in Campo Grande, MS – Brazil. They develop and maintain software for large telecommunication companies, working in close collaboration with their customers. They have four software development/maintenance teams, three of which participated in the study. Although Team A1 and Team A2 share the team manager and one senior developer, they have one dedicated team leader each. The team leaders also act as product owner and software analyst for their teams. Team A3 has a different organizational structure, with one team manager who also plays the team leader role. In A3, the software analyst/product owner is not the team leader. Additionally, all teams are composed of software developers and software testers.

The company adopted hybrid development methodologies, which is the norm in the software industry [218]. Several of their practices come from Scrum, like sprint planning, stand-up meetings, and sprint reviews—but they did not apply Scrum by the book. For instance, regarding roles, the teams had specialists and were not cross-functional. Regarding practices, tasks were mostly self-assigned, but we also observed the team leader assigning tasks. Masood, Hoda, and Blincoe [219] have reported these and other variations as part of Scrum in practice, emphasizing that some are not necessarily a misuse or abuse of the method.

Although Planning Poker is the most used estimation technique in agile software development [33], Teams A1 and A2 abandoned it after trying for a while. Their software teams were inexperienced and young, leading to slow justification rounds and long estimation sessions when using Planning Poker. They switched their technique. Their estimation sessions have the team leader's presence and at least one estimator representing team members for each of their primary software development activities: backend development, frontend development, and software testing. The estimators are not necessarily the team members allocated for the estimated tasks, although there is a high probability that they are. As we illustrate in Figure 4.1, the estimation sessions encompass two main steps: gathering individual estimates from the estimators responsible for each activity in software development (Step 1) and deciding the team's internal estimate (Step 2), which is the value that the team commits to with the team leader. After the estimation session, the team leader converts the internal estimate to the final estimate, defining the team's commitments with the customer (Step 3 in Figure 4.1).

Team A3 still uses Planning Poker for estimating. Interestingly, they report they also want to change their method for the same reasons Teams A1 and A2 did. All teams describe their user stories and tasks in cards and make them available to the whole team on Jira. Teams A1 and A2 also maintain a physical board for their tasks, updated daily. Additionally, all teams carry out the estimation of items and tasks immediately before the beginning of their sprints.

In the second phase of the study, we invited software professionals from other companies to understand our results in different contexts, thus shifting the unit of analysis from companies to individuals. We interviewed four practitioners from four other companies. We expected that this would either confirm the results we found so far or lead us to discover more aspects about them. We selected such participants because their teams conduct software estimation activities regularly. All the participants reported that their companies use agile practices. All of them rely on expert-judgment for estimation, while none adopt Planning Poker.

These participants are from two companies, A, B, C, and D. The participants from companies B and C were from Manaus, AM—Brazil. Company B develops software for a large multinational electronics company and employs over 1,000 people. Company C is a medium-sized company, with around 100 employees in total. They develop a wide variety of software solutions for companies in the electronics business, telecommunications, car dealerships, and others. Companies A, B, and C have in common the fact that they develop customized software solutions for specific companies.

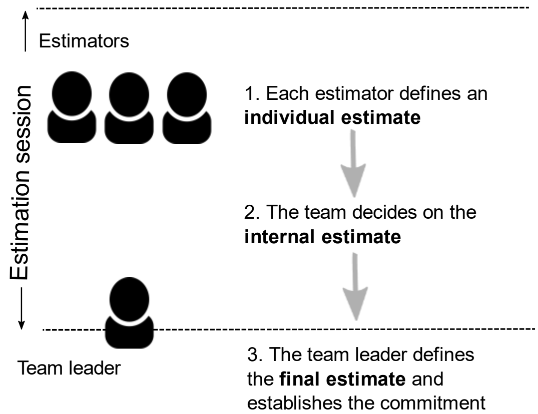


Figure 4.1 - Estimation session.

Companies D and E develop subscription-based software. Their estimation process revolves around the launching of new functionality/products. Company D is a large company in North Vancouver, BC—Canada, with 1,200 employees, developing software for the real estate business. Company E is in Curitiba, PR—Brazil, providing solutions for the financial area. It is a large-sized company, with around 700 employees.

4.2.3. Data Collection and Analysis

Our study was conducted in three sequential rounds of data collection intertwined with data analysis: two rounds in Phase 1, focused on one company, and one round in Phase 2, expanding the research to other companies. In each round, we collected and analyzed data, using the output from one round to inform the next one's design, as Figure 4.2 shows.

In the first round of data collection and analysis, we observed software estimation sessions and stand-up meetings[13] in Company A, as Table 4.1 shows. One of the researchers spent half-days in the company for 38 days from January to March of 2020. In total, this represents approximately 152 hours. The researcher was available to participate in their

[13] We included the guiding questions for observation here: https://www.doi.org/10.6084/m9.figshare.13105319

activities whenever there was the opportunity to do so. The researcher participated in all the estimation sessions during these days, taking notes related to the research questions.

Table 4.1 - Observation sessions.

Team	Observation sessions	Participants
A1	Three estimation sessions	Team leader, software developer, and software tester
A1	20 daily stand-up meetings	All A1 team
A2	One estimation session	Team leader, software developer, and software tester
A2	20 daily stand-up meetings	All A2 team

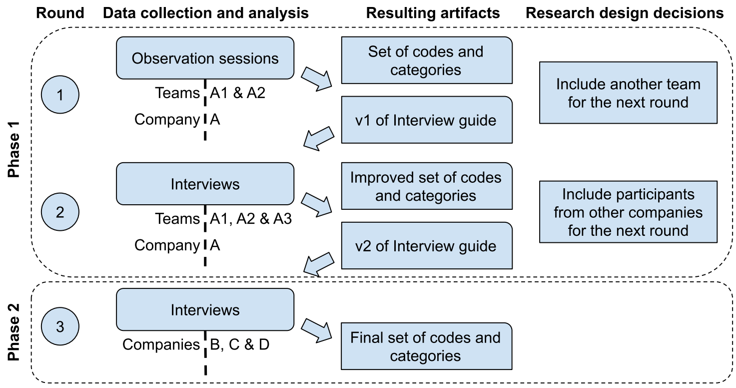


Figure 4.2 - Data collection overview.

After observing the first estimation session, we analyzed the collected data, focusing on open coding **[220]**. We then proceeded with the other observation sessions. Again, we conducted open coding procedure to analyze the data. This round also resulted in an interview guide based on the main results from the analysis of the data from the observation sessions. For instance, we noticed that padding was a recurring theme in estimation sessions. Therefore, we included the following question in our interview guide:**[14]** "*In which situations do you add padding to software estimates?*"

In the second round, we interviewed Team A1, A2, and A3 members, as shown in Table 4.2, resulting in improvements to our set of codes and categories. We also changed our interview questions to focus more on disagreement resolution issues during estimating and on padding due to the analysis we had up to that moment. After the second round of data collection and analysis, we decided to investigate other contexts, moving to Phase 2 of the study. We proceeded to a third-round collecting data, in which we interviewed software professionals from the other four companies—represented in Table 4.2 as Companies B, C, D, and E.

[14] We included the interview scripts here: https://www.doi.org/10.6084/m9.figshare.13105319

Table 4.2 - Interviews.

Interviewees	Roles	Company
P1, P2	Team Manager	A
P3, P4, P5	Team Leader, Product Owner, Software Analyst	A
P6, P7, P8	Software Developer	A
P9	Team Leader	B
P10	Business Analyst, Software Analyst	C
P11	Software Developer	D
P12	Software Developer	E

One of the researchers interviewed the participants, taking notes of their answers. At this point in the research, we intertwined data collection and analysis even more by coding each interview before proceeding to the next one.

During the data analysis, we created codes associated with the relevant parts of the annotations. The researchers held meetings to reach a consensus about the codes and ensuring they were grounded on data. We applied constant comparison throughout the analysis leading to the continuing evolution of the set of codes. We also discussed the relationships between the codes during the meetings as part of axial coding **[220]**.

Finally, we presented our research results to participants of Teams A1, A2, and A3 in two meetings. They considered that the results were correct and reflected their current practice.

4.3. Results

This section presents our research findings regarding ***RQ*** – ***How are software estimates used to establish software development commitments?*** We present the codes and categories[15]—these last ones in bold—starting with the phenomenon of **defensible estimates**. Next, we move our attention to the **padding phenomenon**, a central part of converting software estimates into commitments. We explore the **padding scenarios** and **the reasons to pad**.

4.3.1. Defensible Estimates

Since the teams we observed decided to abandon Planning Poker, their estimation sessions start with each participant providing their individual estimates. Following, the team defines their internal estimate. If they all agree on the individual estimates, then the value is set. However, they may face disagreements, leading them to adopt disagreement resolution

[15] See more of our categories, codes and supporting quotes https://www.doi.org/10.6084/m9.figshare.13105319

strategies. For instance, the estimator may justify the given individual estimate as one step towards disagreement resolution, even though it may not be enough. If everyone accepts the justification, the proposed individual value is accepted. However, if they reject the justification, another step is to do another estimation session later. We also observed that when a disagreement occurs, they might set the estimate as an average of the proposed individual values or as the highest individual estimate.

Finally, another occurrence we observed in the face of disagreements is that the estimator changes their individual estimate. These changes happen in two situations: (i) when the other estimators strongly disagree with an individual estimate or (ii) when the team leader expresses that the internal estimate value is not defensible, i.e, other stakeholders will not accept it. We illustrate this case with the following excerpt of an estimation session. In any case, the estimator moves to a more optimistic estimate.

> OP16[16]: "I believe it takes three to four days to full development because for each period of the day developing for the web, I take two periods developing for the mobile platform. (...)
>
> OP17, regarding software testing: "It takes four days in total. Two days for local testing and two days for beta testing, because we have to evaluate the impact on System Y[17]".
>
> P4 made a totalization, registering it would take five days for backend development, plus five days for frontend development (in parallel with the backend), plus four days for testing – therefore, nine days in total;
>
> P4: "I'll wait for the confirmation of the frontend development estimate. But you have to give it to me today." Next, thinking aloud: "But I don't know whether I can defend nine days..."
>
> So, OP17 answered that it could be one day and a half for each test type.
>
> Then P4 said: "well… I can defend for eight days!"
>
> **Estimation session from Team A1**

The team leader considered the internal estimate not defensible at first. Then, one of the estimators changed his position—his individual estimate—to a more optimistic one to help the team leader to get to a defensible estimate. This occurrence led us to ask the team leader what makes an estimate defensible: "*Some estimates that software developers give me have too much*

[16] OP stands for observation participant.
[17] Names are omitted due to confidentiality issues.

padding, then I don't buy it. So, if they don't convince me, I won't be able to defend it. If they explain it to me during the estimation session and it makes sense, I accept it." (P4, Team Leader). Therefore, when the individual estimate is too high, it is not defensible. The team leader has some notion about the task complexity because the senior developer and the team manager give a baseline estimate for the task before the estimation session.

However, individual estimates with explainable padding are acceptable, and discussing the solution options also contributes to the defense. The degree of novelty and the complexity are the task characteristics that explain an estimate, making it defensible, as the team leader continued to explain: "*So, a defensible estimate to me has a lot to do with the task complexity. I ask myself: are there many business rules involved in this task? Is there anything like this we have done before? If it is too novel or difficult, we must understand it and build the logic behind it with the team to inform the scope description*." (P4, Team Leader).

To lower the pressure while maintaining a good relationship with the customer, the team leader devised a strategy to stand for the final estimates: detailing the items that make up the task and the estimate. P4 (team leader) talks about this: "*My customer is highly resistant to the deadlines I give. He tries to shorten all of them. The way I found to deal with this is to detail all the items of the estimate. This strategy is becoming our standard one, especially when the estimate is a little high because then the customer has no arguments. (...) A few days ago, I had registered a task on our tool, and the customer called me to talk about the deadline. When I informed him, he was like: "What"?!" In these situations, I must explain to him the estimate, showing item by item as I have registered in the tool, confronting them with the scope description (...) And I inform the deadline for each item. For more complicated tasks, I refine even further. So, the customer is accepting the deadlines I tell him. And this pressure is highly contingent on the customer*".

Takeaway message 1: Apart from getting consensus from the team and making estimators committed to their estimates, estimation sessions also focus on building a defensible estimate.

4.3.2. Padding Scenarios

Our results suggest that the padding phenomenon is essential in establishment of commitments with the customer. We identified three scenarios for the interaction between estimate and padding: (i) the estimator embeds padding in the individual estimate; (ii) the team leader adds padding to the internal estimate; and (iii) no padding at all is added to the estimate.

The first scenario is when the estimator embeds padding in the individual estimate. In this case, the estimator pads as part of giving their individual estimate. About this, P11 (software developer) states that: “Every developer has their own estimation method. I believe we all add padding internally, but no one talks about it. As I am an optimistic fellow, I always pad, but I don’t talk about it. If I think it takes one day, I will say it takes three. Some people may do it for slacking, but I do it because of my optimism since I have already had trouble giving lower estimates. Especially at the beginning of projects.” P4 (team leader) also talked about it in one interview: “*The software developer wants to work without pressure. I receive lots of estimates with padding from them. It is rare to get an estimate of something to be done in one hour*.”

The second scenario is when the team leader adds padding to the internal estimate—i.e., to the estimate the team collectively agreed on—before committing with the customer. The following quote from P3 (team leader) shows it: “*So I take the team’s estimates, and I add some padding—one or two days if the task is small and up to five days if the task is large—because the team is too inexperienced. (...) I always consider whether the person giving me the estimate is more optimistic or pessimistic. (...) In my team, we have a super optimistic fellow. So, we need to add more padding before giving the customer the estimate*.”

Also, the interviews with team leaders revealed that the presentation of estimates during the establishment of commitments requires care. Uncertain estimates are interpreted as single-point estimates. Approximate values are interpreted as padded estimates—and the customer tends to reject them. P4 explains it: “*Also, if I tell my customer the estimate is around fifteen days, he will assume it is exactly 15 days. P1 told me that in functionality Z he informed an estimate in the “around of” format, and it became a commitment.” P3 also explains more about it: “In this process, we realized that if we Inform the customer deadlines with round numbers – like 10, 20 or 30 days – he always complains the value is high because he suspects we rounded the number. So, if the software developer said it takes 15 days, we inform the client it will take 17 days, for instance*.”

Another important finding of our study regarding such a scenario is **padding awareness**: the team leader sometimes conceals from the estimator the padding in the final estimate. P4 (team leader) discusses it: “*If the team tells me they are spending six days on it, I say we will spend more–- within acceptable limits. And I do not tell the developers of the padding I added”. One software developer (P8) also revealed more about how he suspects the team leader adds padding, but software developers are not aware of it: “I believe P3 [the team* *leader] adds padding later, but it is not of our concern. (...) During the meetings, they told us that the padding is to raise the confidence of the team leader with the customer*."

Therefore, when the team leaders add padding to the internal estimates, they may be trying to raise the confidence that the team will keep commitments with the customer. However, they conceal the padding they added from the estimators, making estimators accountable for their individual estimate if the task is given to them. P8 (software developer) reinforces this: "*They [team leaders] don't convert the padding to the team—at least it is what they said in the meeting. If the developer estimates five days during the estimation session, he has five days to finish the task.*" Other reasons for concealing the padding from estimators are that high estimates may give the impression there is plenty of time to execute the task and lead to lower productivity.

However, there are situations when the estimators are aware of the padding the team leader adds. The transcription below shows an observation session where the team leader first says there is no room for padding for software development activities, but later during the meeting adds padding to the software testing activity.

> OP13 listed all the classes he remembered and concluded that the task would take at least one ideal day of work, depending on the person who will execute the task; OP14 agreed with him;
>
> OP15 said that in his opinion, it would take two days; OP13 reaffirmed that his estimate was contingent on the person executing the task;
>
> OP14 commented that they always pad a little. However, P3 said that this week there is no room for padding.
>
> OP20 estimated one day for testing. P3 said he would count one day and a half to test because of other stuff, which is also necessary to verify.
>
> In the end, the estimate for development only was one day and a half.
>
> **Estimation session from Team A2**

Overall, estimators are aware of padding when a similar task is complex, the task involves problematic parts of the system, or there is a need for more robust testing. In the specific case of the abovementioned estimation session, the team leader needs the padding to ensure that they will have enough tests – and the team leader makes it straightforward for the team that this is how they are going to use the padding. In doing so, the team leader is limiting the use of padding.

The third scenario is when no padding at all is added to the estimate. Different reasons cause this scenario during the establishment of the commitments with the customer: when the task is urgent, the task is simple, the task is noticeably clear to managers, or when the task has a pre-defined deadline. Additionally, when the customer has a high expectation over the task, it is also impossible to pad. The same happens when the task seems simple, although it is not, and the customer has some technical background. In one of the companies, their context requires technical expertise, which may not be available in their team. In this case, no padding is possible when there are qualified personnel to execute the task.

P3 (team leader) mentioned several of these reasons when asked about when padding is impossible: "*In situations where an outsider would consider the card to be simple, but the implementation is like "may God help us." For instance, module ’'s code is quite tricky because there is an impact in many other parts of the system. However, from the perspective of the customer, it's simple. (...) In urgent situations, it is also impossible to pad. Also, in tasks in which the customer has high expectations, we cannot make late deliveries. Those are the cards that lead us to overtime work. Another case is when the task is a promise from our board of directors. We receive them closed, with a defined deadline. (...) We also consider who the customer is because sometimes he has a technical background and will not accept padding, depending on the task. If the task is about labeling a field, he won't accept padding at all.*" Therefore, as the different padding scenarios show, there are tasks for which padding is not viable.

Takeaway message 2: The use of padding varies across three scenarios: (i) the estimator embeds padding in the individual estimate; (ii) the team leader adds padding to the internal estimate; and (iii) no padding at all is added to the estimate.

4.3.3. Reasons to Pad

Our data also revealed that there are mainly three different reasons to pad: (i) padding for contingency buffer, (ii) padding for completing other tasks, and (iii) padding for improving the overall quality. We give an example of each of these reasons in Figure 4.3, which we explain in the next paragraphs. The example is a simplification of reality since a real task's padding may involve all three reasons to pad. Our illustration makes a didactic separation of each reason. It also includes the example of a task with no padding at all—Task A—to aid in the development of our argument for one of the reasons to pad.

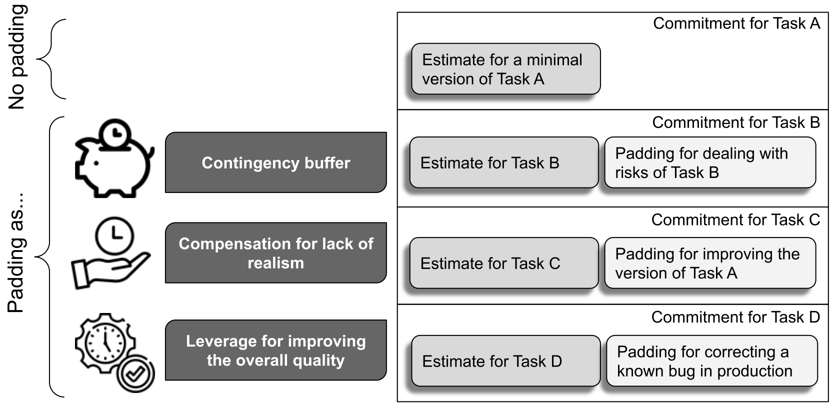


Figure 4.3–- Reasons to pad.

First, team leaders and estimators may use padding for contingency buffer to deal with risks that may occur during software development and maintenance of a task and raise the chances to fulfill deadline commitments. We illustrate this reason to pad in the case of Task B in Figure 4.3, where padding was added to Task B’s estimates to keep a reserve to deal with risks associated with this task. P8 (software developer) discusses it when questioned about why software developers give higher estimate values: “*Usually, it is because we are afraid of the problems we will have to face. Like, in larger tasks or tasks that involve implementation in some specific parts of the system, which have higher chances to have a problem there*.” The estimator also adds padding to their individual estimate, fearing accountability due to delays. It may also be the case that the estimator considers himself optimistic, or even because the task depends on another one executed by a teammate known to make deliveries with errors. Padding individual estimates may also happen when estimators have too many doubts regarding the task features, leading senior developers to consider problems during development.

Estimators also pad their individual estimates for more technical reasons, like the lack of familiarity with the company’s code, if they have just begun a new job, as P11 (software developer) discusses: “*When I am at a new company, as I am not familiarized with their code, I add a high value of padding*.” Alternatively, they add padding when there are dependencies among tasks demanding lots of communication. Another more technical reason is when the task is related to problematic system modules.

Padding for contingency buffer is also useful when the team leader adds padding to the internal estimate when generating the final estimate. It may be used due to the tasks’ characteristics, like when the task is large, critical, complex, or ill-defined. P9 (team leader) talks about this when asked in which situations he pads estimates: “*When we cannot define the* *feature very well. We need to carry our feasibility studies, but there is no time to do it because it is time to make a proposal*". It also happens when the implementation requires a learning curve or there is no time to investigate more about the task. Additionally, the team leader pads the internal estimate due to experience issues, like the team leader's past experiences or when the team is inexperienced. More technical issues may also play a role in padding for contingency buffer since it may happen because there is a need for integration with third-party software or the lack of technology specialists.

Specific issues related to the estimator's characteristics may also cause the team leader to add padding to the internal estimate, like when the estimator is inexperienced at the company. Another reason is a known higher level of optimism of the estimator, a known higher level of deliveries with errors from the estimator, or when the estimator is insecure. P3 (team leader) talks about it: "*Nowadays, I know when a task is going to return [with errors from the test] due to the experience I have with the person [assigned to the task]—then we add more padding. If the person is optimistic, we also pad.*" More generically, the team leader may pad for contingency buffer simply to deal with unforeseen problems or raise the confidence that the team will meet commitments. In any of these cases, dealing with risks seems to be essential.

The second reason for padding is padding for completing other tasks. It happens to gain time to implement a task that estimators or team leaders could not add padding for. We illustrate this case with Task A and Task C in Figure 4.3. There was a need for padding to ensure the complete implementation of Task A, but the context did not allow for it. Therefore, the decision was to estimate Task A to deliver a minimally viable version of it and pad the Task C estimate to compensate. The padding of Task C is meant to be used to finish the full version of Task A instead of being used for Task C implementation. P3 (team leader) talked about this: "*We may use padding to gain time for a task that we could not add padding (...) We gave an estimate of 30 working days for functionality Y, but we are counting on the padding of other tasks to finish it*." Notice this reason to pad connects with the scenario where there is no padding added to the estimate.

Padding for completing other tasks also happens when estimators or team leaders use padding from one task to implement tasks planned to, but not delivered in previous projects/iterations. P10 (business/software analyst) gave an illustrative example of this: "*Sometimes, we have a contract including functionalities A, B, C, and D, but we do not deliver D, for instance. So, we will implement D in another project, which includes other requirements, and we add padding for D*". In this case, the need for completing is also there, but the granularity is larger: the team needs to complete an entire project instead of a single module or functionality.

The third reason is padding for improving the overall quality of the product. It happens when team leaders use the value of padding when there is a need for more robust testing or to implement tests. It also happens when the team wishes to implement improvements in the system, or simply to develop carefully. Another motivation is to allow for the correction of bugs in production, as we illustrate in Task D of Figure 4.3. Finally, padding for improving the overall quality may also happen to evolve well-accepted features. In other words, the estimator or the team leader can use the value added to the internal estimate to ensure the fulfillment of the established commitments with the customer, including overall quality commitments. P3 (team leader) talked about it in the interview: "*It also happens that there are errors we know, and we add one day in one task to correct it. For instance, we delivered functionality T, but we were not able to test it. After we delivered, the testers started to work, and they found lots of bugs. Now we are correcting these bugs.*"

On the one hand, it looks like team leaders use padding to meet short-term commitments, either by not padding tasks that, for instance, the client has high expectations or by including padding in estimates of tasks at hand to deal with risks. On the other hand, they also use padding to keep up longer-term commitments, like when they compensate for the lack of realism in some tasks through others' padding. The lack of time to dedicate to quality requirements in some tasks is also compensated through other tasks' padding, leveraging the produc''s overall quality. Therefore, they make sure to execute all tasks and satisfy all commitments in the long run.

Takeaway message 3: We found three reasons for padding estimates: contingency buffer, completing other tasks, or improving the overall quality. These different uses of padding emerge from the estimation process to ensure both short and long-term commitments can be met in the long run, even when they are conflicting at a given moment of software development.

4.4. Discussion

In this section, we discuss our results considering the existing literature. We start with the finding of defensible estimates, in Section 4.4.1. Next, we move to the finding of padding in Section 4.4.2.

4.4.1. Getting Defensible Estimates

One of our core findings is that estimation sessions serve to build defensible estimates, in addition to getting consensus among the team members and their commitment to estimates, as we presented in Section 4.3.1. This happens because team leaders may not be willing to accept estimates they are not convinced of. After all, they are the ones in contact with customers and who will negotiate with them to establish commitments based on these estimates. In response to this, estimators may explicitly change their initial estimates to more optimistic ones if their team leaders do not consider them defensible. However, changing estimates to more optimistic ones may lead to unrealistic estimates and errors.

The finding of the changes to estimates also aligns with the ones from an interview study with large and mature organizations, where Magazinius et al. [7] report that estimators may decrease estimates due to management pressure or may change estimates to attain to organizational agenda, like due to the interests of customers. These results show that, instead of standing up for their estimates and treating them as non-negotiable facts, technical staff still need to learn skills to convince their bosses of their estimates, as McConnell **[221]** said they were years ago. Thereby, it is not enough to provide estimators suitable methods for reaching consensus over an estimate or for generating a realistic estimate: they also need methods to aid the defense of their software estimates.

Implications for practice: To avoid pressure over their estimates, software professionals need to help their team leaders and managers to build up arguments for defending estimates during the establishment of commitments.

Implications for research: Our results indicate that estimators change their estimates to more optimistic ones under pressure. Therefore, we need practices that empower such estimators to defend their estimates to keep them realistic.

4.4.2. Padding to Buy Time

Another finding of our study concerns the padding phenomenon that we explored in Sections 4.3.2 and 4.3.3, which involves adding a value to the original estimates before their communication when defining a commitment. We found industry scenarios in which padding is impossible, even if the team feels it is needed. When padding is viable, our findings indicate it is used to “buy” time for three reasons: for contingency buffer, completing other tasks, or improving the overall product quality.

Padding for contingency protects against risks in software development, buying time to deal with them. The use of contingency reserves for schedule and budget is already known as a recommendation for project management in the Project Management Body of Knowledge (PMBOK) [74], a good risk management practice to fight against fires that may impair a software project [222], and as a mechanism to compensate for the winner`s curse [26]. Also, the inclusion of a large buffer to deal with unexpected events or changes in specifications is a reason for accurate estimates [96]. Additionally, Magazinius et al. report that project stakeholders from industry sometimes intentionally increase their estimates to avoid overspending software development resources [7].

Therefore, the use of padding for contingency buffer is valid and vital for software tasks' execution—and it has been widely recognized in the software engineering literature. However, our results indicate that software professionals use padding for two additional reasons: completing other tasks and improving the overall quality. For completing other tasks, padding is added to one task to gain time to implement another one that they could not add padding for. The last task's commitment is not realistic, and the padding of other tasks counterbalances this fact. It also happens when a task was planned to be delivered in a given project/iteration but is not. Then, padding may be added to other projects/iterations to include these tasks. In any of these cases, padding serves for buying time for those other tasks.

Another reason to pad is for improving the overall product quality by implementing improvements in the system, amplifying tests, or allowing for the correction of bugs in production. It is like buying time to attain to quality requirements, satisfying long-run commitments. In alignment with our findings of padding for completing other tasks and improving the overall product quality, Magazinius et al. [7] report that the most common reason for intentional increases of estimates in their study was for hiding other activities in the estimated ones. They state this happens either to get more development time for one functionality or other testing or maintenance activities [7].

In such cases, padding is a managerial mechanism to allow for the repayment of technical debt in the software products. Lim, Taksande, and Seaman [223] report that management may not always recognize the importance of repaying technical debt unless they are rewarded or the customer is willing to pay for it. Additionally, customers may not be willing to give software teams the time to repay technical debt unless they get direct value from this [223]. Our findings indicate that in such a scenario, using padding to implement tasks not delivered in previous projects–- padding for completing other tasks—is a way to repay requirements debt. Also, padding for implementing tests, implementing improvements in the system, or allowing for the correction of bugs in production–- padding for improving the overall quality—is a mechanism to repay design, coding, or testing debts. Therefore, while researchers are focusing on more technical approaches for repaying technical debt, like refactoring, rewriting, automation, and others [224], industry professionals also have to find managerial paths to allow for such repayments, like padding their estimates.

Additionally, Becker, Walker, and McCord [225] mapped studies about intertemporal choices – a concept of psychology and behavioral economics referring to "*decisions involving tradeoffs among costs and benefits occurring in different times*" [226]—in software engineering. They found that no empirical work investigated trade-offs in time in depth. Our study contributes to filling in this gap, providing evidence about how practitioners use padding—or the lack of it—to balance short and long-term needs. Customers may have a strong focus on shorter time to delivery and lower costs, leading teams to sacrifice quality during software development. Such an attitude reflects on the estimation process, and the set of tasks at hand in a particular moment receives much attention. In this context, padding is a mechanism that team leaders and managers use for buying time to deal with risks in software development, to compensate for the lack of realism of previous tasks, or to improve the overall quality of the product in the long run.

Along with our results, the findings from these other studies indicate that padding is a relevant practice in the industry's estimating process, especially for protecting software projects from risks and providing managerial mechanisms for the repayment of technical debt. An interesting remark is that one of the team managers asked the researchers to present the research results regarding padding to a novice team leader for training purposes, which indicates their practical usefulness and relevance. It is time to recognize padding as another tool in the software engineers' toolbox to deal with estimation's social and human aspects.

Implications for practice: Padding is a relevant practice in the software engineering's toolbox and goes beyond providing a contingency buffer: it is also used to complete other tasks and improve the overall quality of a product. Practitioners can use our results to train novice team leaders on when and why to pad, given the reasons we found.

Software teams can also classify their padding of tasks according to the reasons to pad. Too many tasks with padding for completing other tasks or improving overall quality suggest a need for improving—or perhaps defending—estimates.

Implications for research: Padding hides the balancing of short and long-term commitments from customers. Sometimes a task is not padded to satisfy a short-term need—like delivering faster—but another one is padded to compensate for the resulting lower quality—like for correcting bugs left due to the absence of time for testing correctly. A better comprehension of padding in the software industry aids researchers in proposing alternative or supplementary practices to padding to make the balance of short and long-term commitments more transparent and controllable, instead of just yielding to the pressure of short-term needs.

4.5. Limitations

One of the limitations was that respondents might have understood interview questions differently from what we meant. To minimize this, we executed the observation sessions before the interviews to ensure we would use participants' terminology. By doing so, we also focused on specific behaviors closely related to our research questions.

Also, there was the risk some topics were too sensitive for participants to mention, as the changes of estimates and padding behaviors. For instance, in the research about distortions of software estimates, Magazinius et al. [7] comment on how some of their respondents asked them to stop audio recording in some parts of the interview to inform about sensitive issues. Likewise, we were running the risk of having our results biased by political reasons. We mitigated this risk by being in constant contact with the team and executing observation sessions for an extended period, making it unlikely that sensitive behaviors would be covered. We intended to promote an environment where participants could speak freely about any subject, including sensitive topics. Therefore, we did not audio record the observation sessions and interviews, raising the risk for misunderstandings. We showed a sample of our annotations to the participants of the observation sessions to validate them, to minimize this threat. After each interview session, we typed all the annotations and emailed them to the interviewee, asking him/her to read them and point inaccuracies. Additionally, we presented our results to some participants to assess their resonance—and we received positive feedback.

We first analyzed the data from one single company. Therefore, it was hard to say our results generalize to other contexts. We interviewed software professionals from other companies located in other cities and working in different business areas to address this. Despite the variation, all companies embrace agile or hybrid development to some extent. So, it may be the case that our results are especially relevant for this context. In any case, our main concern was with understanding a specific phenomenon over having generalizable results.

Concerning reliability, all the researchers held meetings for reaching a consensus during coding, ensuring that the codes were meaningful, representing the quotations and that the relationships between the codes were grounded on data.

4.6. Summary

This chapter presented a study in five companies to investigate the interaction between software estimation and the establishment of commitments with customers. In this sense, our study contributes to untangling the underlying phenomena of defensible estimates and padding, showing how the practices software practitioners use in the field help them deal with the human and social context in which estimation is embedded. First, our results show that the interaction of estimation and the establishment of commitments lead to estimation sessions that focus on more than solely getting consensus among team members and making estimators committed to estimates. It also serves to build defensible estimates, in the sense that these estimates are aligned with customers' and higher managers' expectations. Therefore, instead of defending their estimates, estimators change their estimates to more optimistic ones if there is a belief that they are not defensible.

Second, padding is a valid mechanism in the industry, and team leaders have different reasons to pad. They may use padding for contingency buffer, completing other tasks or improving the overall quality of the product. As a contingency buffer, padding serves as a reserve to deal with risks during software development. For completing other tasks, the padding of one task embeds the estimates of other tasks for which padding was impossible. For improving quality, padding compensates for previous deliveries where time had higher priority over quality requirements. Interestingly, padding for completing other tasks and for improving quality are managerial paths that industry practitioners have found to repay technical debt.

Therefore, instead of openly defending their estimates to raise their chances to satisfy commitments in the short and in the long-run, estimators pad them. This shows the need to develop mechanisms that support software professionals to defend their estimates, as McConnell [221] suggested years ago. Such mechanisms can be alternatives to padding to help software practitioners to balance short- and long-term commitments in more transparent and controllable ways, whenever possible.

The results of the empirical study that we presented in this chapter complement the findings from our SLM in CHAPTER 3. Together, they show that pressure over expert judgment estimates leads estimators to change their estimates to satisfy customers' expectations or organizational goals, even if this leads to unrealistic commitments. They also show that padding can be used to compensate for that but that higher managers sometimes remove padding, reducing the effectiveness of this tool and leading to estimation inaccuracies and problems (see Sections 3.4.2 and 3.7). We conclude that such results enrich our understanding of our research problem and show its pertinence in our target environment as part of our DSR relevance cycle. Therefore, in CHAPTER 5 we present the result of our DSR design cycle: an artifact to support software practitioners in defending their estimates, instead of changing them due to pressure. This artifact can also be used to defend or to complement padding of software estimates, to raise changes of meeting commitments made with customers or higher managers.

CHAPTER 5 – THE DEFENSE LENSES AND THE DIGITAL SIMULATION

This chapter presents the artifact that we designed as the result of the execution of our design cycle, based on the information we gained during our rigor and relevance cycles. We also refined the artifact after the evaluation from the design cycle.

5.1. Introduction

CHAPTER 3 discusses what the SE research literature presents as the factors affecting expert judgment software estimates. Among a myriad of existing factors, some confirm the existence of deliberate changes of software estimates and the difficulty of estimators to defend their estimates and more realistic commitments: *early estimates*, *price-to-win issues*, *goals and targets*, *pressure*, and *padding*. Other factors impact the satisfaction of commitments, also showing its relevance in the estimation context: *reestimation and revision of estimates*, *changes to requirements or scope* and *project flexibility*. CHAPTER 4 presents the results of a qualitative study in the software industry, highlighting the importance of defensible estimates (that is, estimates that are aligned to customers' and higher managers' expectations). It also brings results regarding the change of estimates due to pressure and reveals that padding is a tool to balance short and long-term needs in this context.

These results connects with what we discussed in Section 2.6: when facing a demand, people can fall into the three-A trap [20]: accommodating, attacking, or avoiding. Estimators can also fall into it when they face pressure to change an estimate or feel pressed to accept an unrealistic commitment—one that their realistic estimate to get the job done does not support. Although the attacking strategy does not seem wise in most working environments, accommodating and avoiding seem likely. After all, if their bosses or clients are pressing them, the chances are that estimators will yield to avoid any damages.

We claim there is a fourth strategy that estimators can try in such situations, illustrated briefly in Figure 5.1. Estimators can defend their estimates, Getting to Yes [18] with their customers and higher managers regarding a realistic one. They can work to Getting Past No [19] from their customers and higher managers to commitments made based on such realistic estimates. In addition, estimators can experience the Power of a Positive No [20] to pressure and unrealistic commitments.

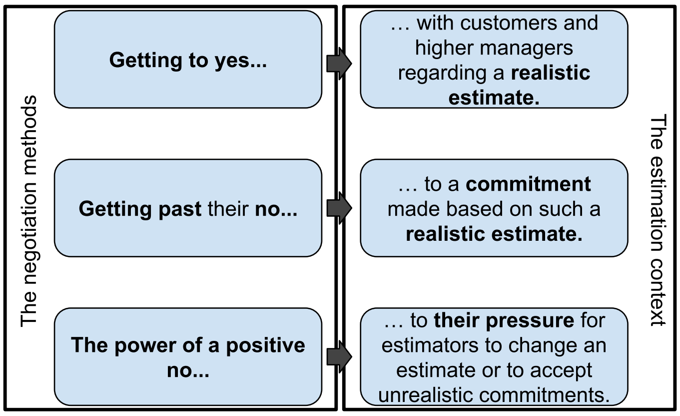


Figure 5.1– The negotiation methods in the estimation context.

Based on this, we created an artifact to support estimators in defending their software estimates. The artifact is not about how to make a technically sound estimate. Assuming that the estimator is confident that they have a good estimate for the task/project and are ready to communicate it to other stakeholders, they can use our proposed artifact to resist pressure from other stakeholders. Moreover, we designed the artifact for individual estimators communicating their estimates (in estimation sessions, for instance), but the approach can be used by software teams communicating a team estimate to stakeholders.

The artifact that we designed in our first design cycle has the format of defense lenses, inspired by the design lenses that Deterding [35] proposed to guide the design of gamified systems. Design lenses combine a name, a design principle, and a set of focusing questions, supporting the designer to take a mental perspective regarding the design issue considering the lens [35]. Likewise, the defense lenses also have a name, a negotiation principle on which we based them, and a set of focusing questions. The idea is to provide estimators with a mindset shift from changing estimates to defending them if no legitimate reasons justify a change. Also, the defense lenses must support estimators to take a mental perspective of negotiating commitments that others are trying to impose when they are unrealistic, instead of accepting them promptly. In Figure 5.2 we present the title of each one of the defense lenses, relating them to the steps and principles of each one of the negotiation methods that we used.

As Figure 5.2, the set of lenses covers all phases, steps, and principles from the original methods. We illustrated the Positive No method through its three major steps, matching the lenses to the "Yes! No. Yes?" structure. We also present the relationship between a special card that we created—the Wildcard—and other cards from the set of lenses. Therefore, we left the relationships between the Wildcard and the methods implicit.

Despite each lens is designed to be used in isolation if needed, we also organized them in three recommended packs: (i) the Minimal Defense Pack, focused on aiding estimators to defend their estimates when facing particular pressure episodes; (ii) the Extended Defense Pack, that complements the first one, to be used when pressure continues even after the estimator has made an initial defense attempt; and (iii) the Strategic Defense Pack, aimed at helping estimators to reach more enduring changes to their environment when pressure is recurrent and overwhelming. In Table 5.1, we present the packs, what specific situations they can be used, and the lenses that compose each of them.

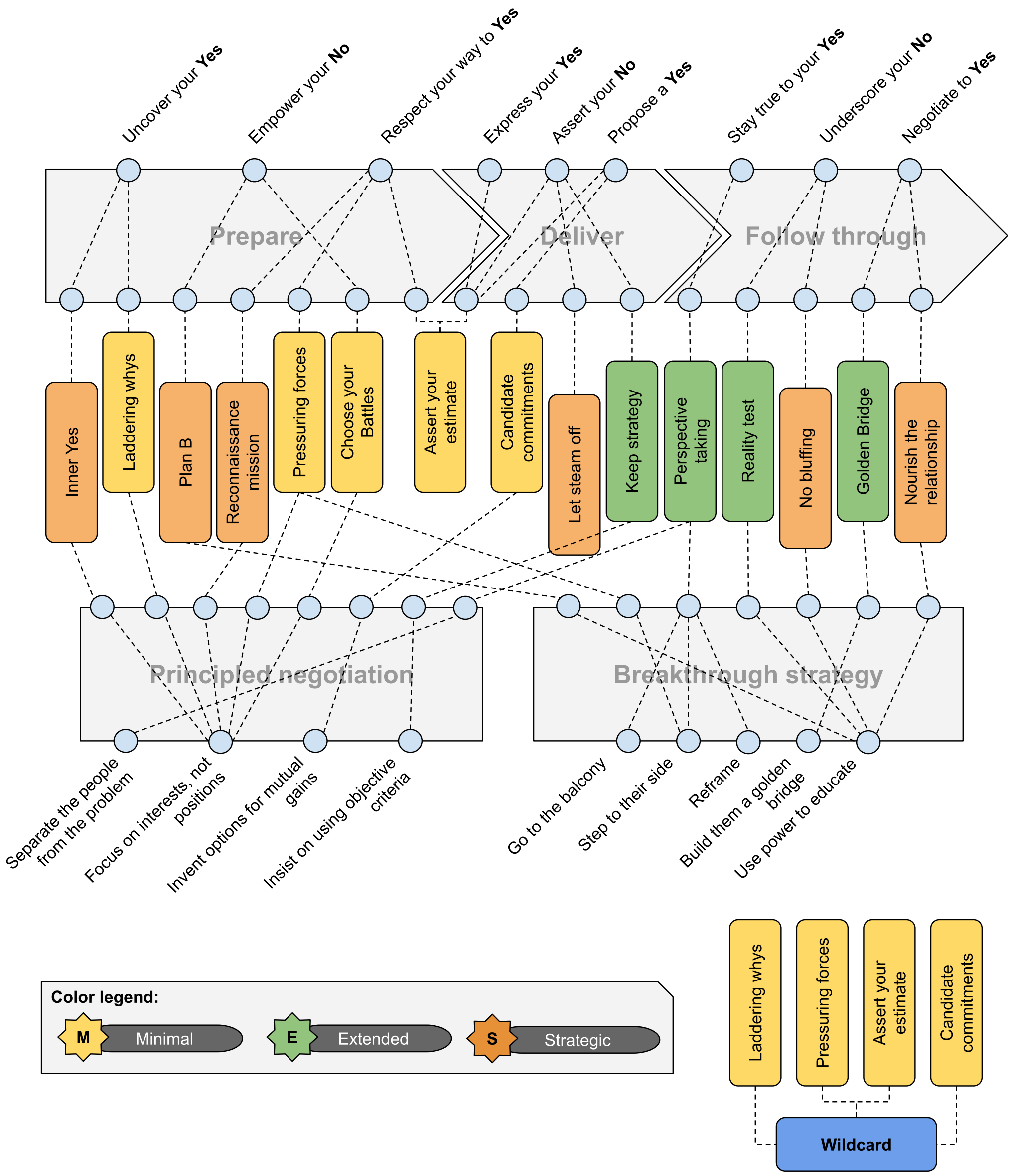


Figure 5.2– The lenses and the negotiation methods.

Table 5.1 – Packs' description.

Pack	What for	Lenses
Minimal Defense Pack	• One is preparing oneself for providing one's estimates and have reasons to believe will face pressure from people receiving them. • One provided one's estimates and is facing pressure to change them. • A client or higher manager wants one to commit to an imposed commitment: fixed deadline, with a closed scope, and with too restricted resources.	• Assert your estimates • Pressuring forces • Laddering whys • Choose your battles • Candidate commitments
Extended Defense Pack	• One tried to deflect from a pressure episode, but people keep pushing for unjustified changes in one's estimates or for the acceptance of an unattainable commitment.	• Keep strategy • Balcony • Reality test • Golden bridge
Strategic Defense Pack	• One or one's team are constantly pushed for lower estimates or for unrealistic imposed commitments, creating an unhealthy environment that one aims to change for good.	• Inner Yes • Reconnaissance mission • Plan B • Let steam off • No bluffing • Nourish the relationship

The packs are useful in different situations, as estimators may not have enough time to prepare themselves with all the steps from the original method. We also included a Wildcard, which is an introductory lens that belongs to no pack, but contains elements of four cards of the Minimal Defense Pack. All the other lenses are, therefore, advanced lenses. Additionally, we described the lenses in the format of cards, and Figure 5.3 presents their schematics. We intended to create a simple-to-consult format while also providing more advanced guidelines.

On their front side, each card has the lens' name and an icon. They also are classified according to their recommended pack (see Section 5.1). Next, we describe the negotiation principle that supports the lens, which helps software professionals and teams to grasp what it is about. At the gray rounded rectangle, we present a set of focusing questions. They aim at conducting the card user to change their perspective in their path to implement the card's negotiation principle during a real-life communication of their estimates. On the backside, the card comprises a handle, which describes the situations to apply that lens. It helps identify the specific lens we need when using them in isolation instead of using the recommended packs.

The card also has advanced guidelines to support less experienced practitioners or help in more complex situations. We present each lens from Sections 5.2 to 5.17.

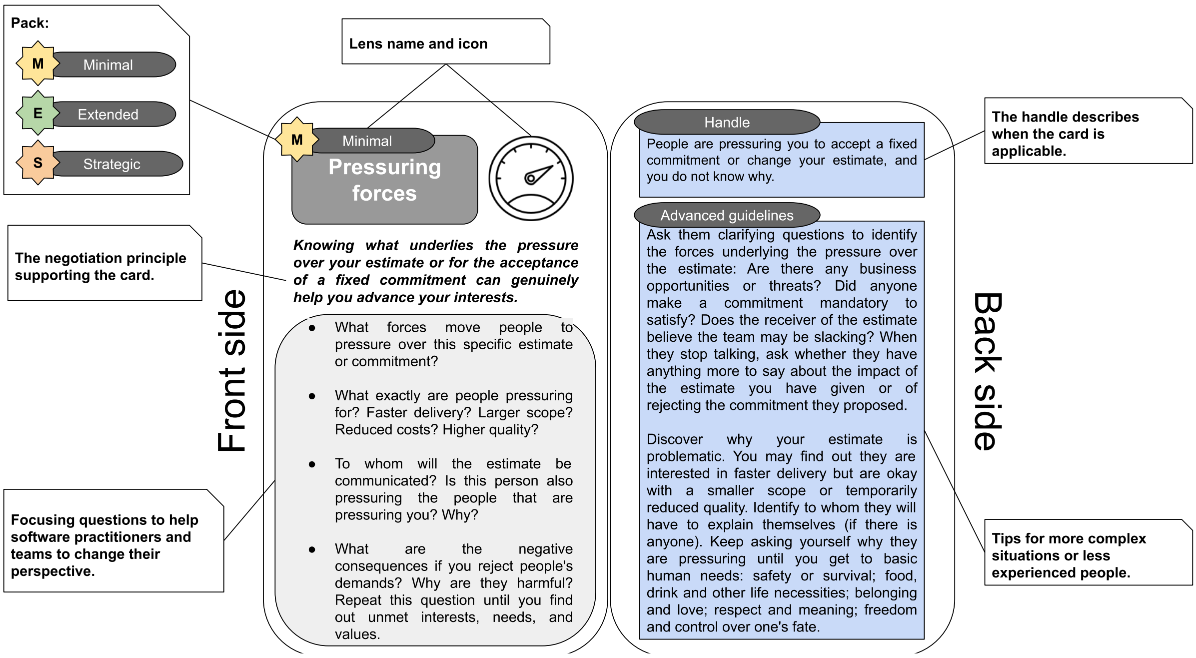


Figure 5.3–- Lens schematics.

To provide a more throughout support to software estimators, we also designed a digital simulation to support the gain of knowledge about the lenses, as part of a second cycle of design in our DSR project. The digital simulation presents two packs: the Minimal and the Extended ones. It has the format of two interactive videos, which we detail in Section 5.18.

5.2. The Wildcard Lens

The **Wildcard** is an introductory lens, aimed at providing estimators with an overview of the main elements of the proposed set of elements. It deviates from the lens schematics presented in Figure 5.3 in two points: (i) it does not belong to a pack, and (ii) we replaced the advanced guidelines section with an advanced lenses section, containing pointers to the other lenses of the Minimal Defense Pack and to the Extended Defense Pack. We present the **Wildcard** in Figure 5.4.

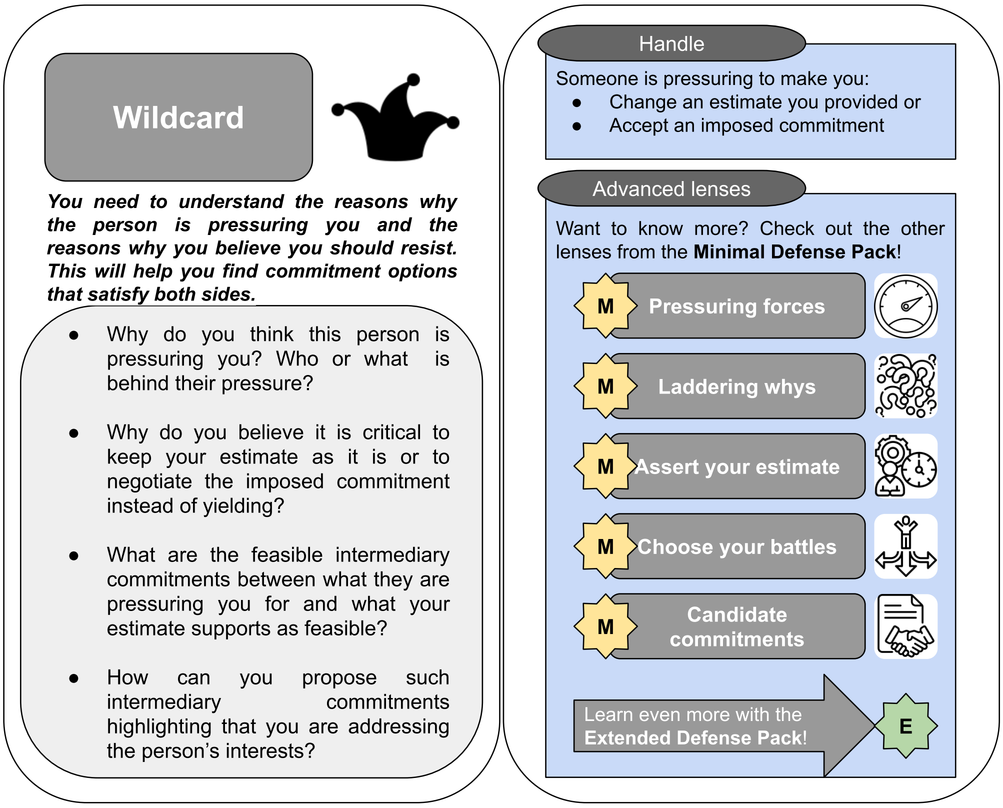


Figure 5.4— The Wildcard.

The **Wildcard**' handle explains its applicability, which is the applicability of the whole set of lenses. Then its four guiding questions are related to four cards of the Minimal Pack: Pressuring Forces, Laddering whys, Candidate commitment, and Assert your estimate lenses. The Wildcard' theoretical foundations are linked to these advanced lenses. We leave its explanation together with each lens of these. Also, the advanced guidelines section was replaced by an advanced lenses section, pointing to its composing five lenses.

Each of the following sections presents each advanced lens individually. We start with the lens from the Minimal Defense Pack, moving on to the Extended Defense Pack, and then to the Strategic Defense Pack. In APPENDIX B we also included a booklet, i.e. a short book, we wrote to the target population of negotiation lenses. It provides a more concise description of the lenses that estimators can use to learn our method or consult it later when needed. In APPENDIX C we included the booklet in Brazilian Portuguese.

5.3. Assert your Estimate Lens

The **Assert your estimate** lens aids in communicating the estimate while ensuring it is in the best interest of everyone. It works by expressing the estimate for avoiding pressure. The lens is based on the following steps from the Positive No Method [20]: Respect your way to Yes, Express your Yes, Propose a Yes, and the Assert your No. Figure 5.5 shows it.

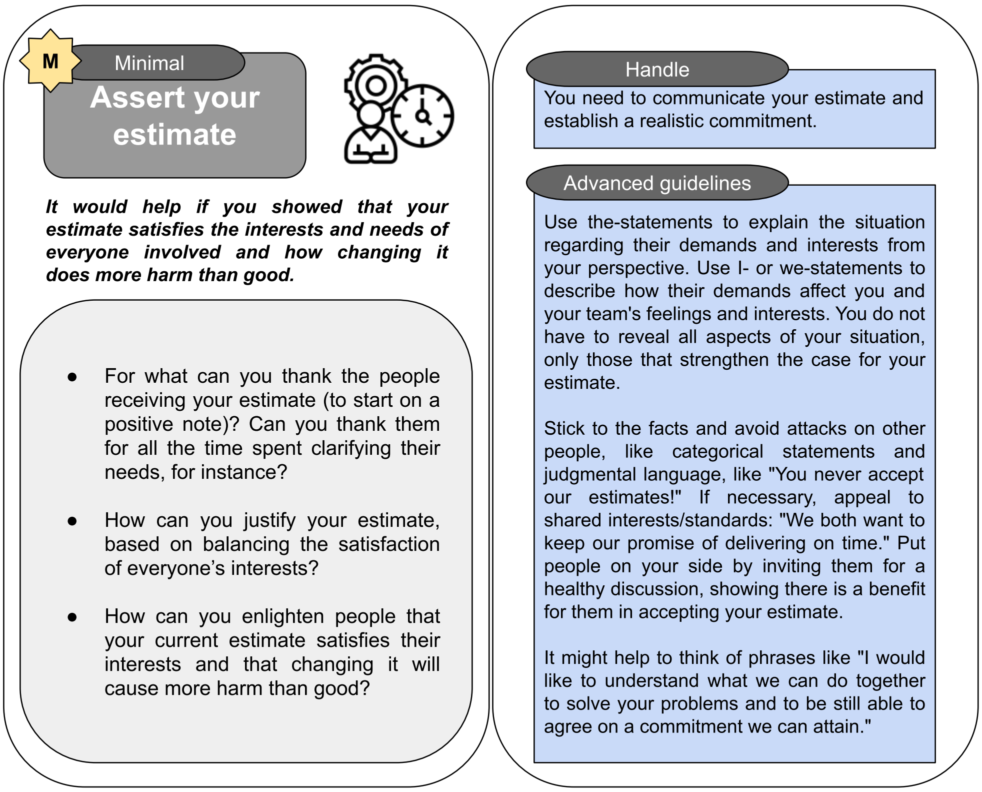


Figure 5.5— Assert your estimate lens.

It might be the case that this is the first lens the estimator is applying, to communicate their estimate and to prevent pressure in the first place. Therefore, this lens proposes that, instead of jumping into the estimate subject, the estimator starts by thanking the other side for something as a means of adopting a positive attitude of respect and acknowledging them. This is especially important if the estimator believes the estimate is likely to be rejected: begin the positive No to pressure on a positive note. So, the first focusing question of the lens is grounded on the Respect your way to Yes Step.

Additionally, the core of this lens is communicating the estimate while assuring that the estimator is thinking on the satisfaction of everyone's interests. When designing this card, we assumed that the estimator knows the overall interests of all stakeholders beforehand: their own, their team's, their manager's, and the clients'[18]. So, the second focusing question requires that the estimator expresses their yes to their interests through the justification of their estimate, a crucial aspect of the Express your Yes Step. The second and third focusing questions also involve proposing an estimate value that respects managers and client'' core interests, a part of

[18] Obviously, estimators may need clarification on interests, depending on the situation. The Pressuring forces and Laddering whys lenses, also part of the Minimal Defense Pack, can guide them in this matter.

the Propose a Yes Step. Estimators are sending their first, subtle message of No to pressure in between the lines of their estimate value and justification. The No is implied, but it is there. Therefore, this lens is also grounded on the Assert your No Step.

We also recommend the use of the-statements, I-statements, and we-statements [20] on the advanced guidelines. They will help the estimator focus on the situation and themselves, instead of focusing on the people pressuring them. This comes from the Express your Yes Step. In Table 5.2, we present the definition and examples of these different types of statements we mention on the advanced guidelines (we also included this table in the booklet to help practitioners).

Table 5.2 – Statement's descriptions and examples.

Statement-type	Description	Examples
The-statement	The speaker describes the situation based on the facts plainly and neutrally, instead of blaming the other side for any problems that exist in such situation (even if they are guilty).	• The deadline we agreed on for the last iteration required the delivery of a less polished feature, and our current estimate makes space for improving it in this iteration. • The implementation of the required feature involves the data migration to a new format, and that takes a lot of effort.
I-statement	The speaker describes their feelings, experience, and interests.	• From my experience, I can tell our team won't make more than the four first items in our backlog in this Sprint without compromising their weekend. • I feel frustrated when there is no room for refactoring in our backlog. • I believe our team is overwhelmed by the amount of overtime work from the last couple of iterations, because the Sprint Backlog included more items than they were capable to handle. Therefore, I think we should leave some room for all the unexpected events we are facing lately.
We-statement	The speaker describes the joint interests of the parties involved, if uncomfortable with describing their own alone. They can also describe fair standards.	• We all want to make sure that the product is delivered on the agreed deadline, right? • I am sure we both want our employees working according to the law.

Although an estimator can use this lens to successfully avoid pressure, it will not be enough in many real-life situations. Therefore, the estimator benefits from knowing all the remaining lenses from our set.

5.4. Pressuring Forces Lens

When facing a pressure event over a specific estimate, or when pressured to accept a predetermined and unrealistic commitment, the estimator can use the **Pressuring forces** lens, that focuses at making the estimator get perspective on what underlies the existing pressure. By using it, they can find out that their estimate represents an obstacle for satisfying a relevant need or interest. Therefore, understanding why people pressure for estimate changes is paramount for establishing a mutually satisfactory commitment. Figure 5.6 presents this lens.

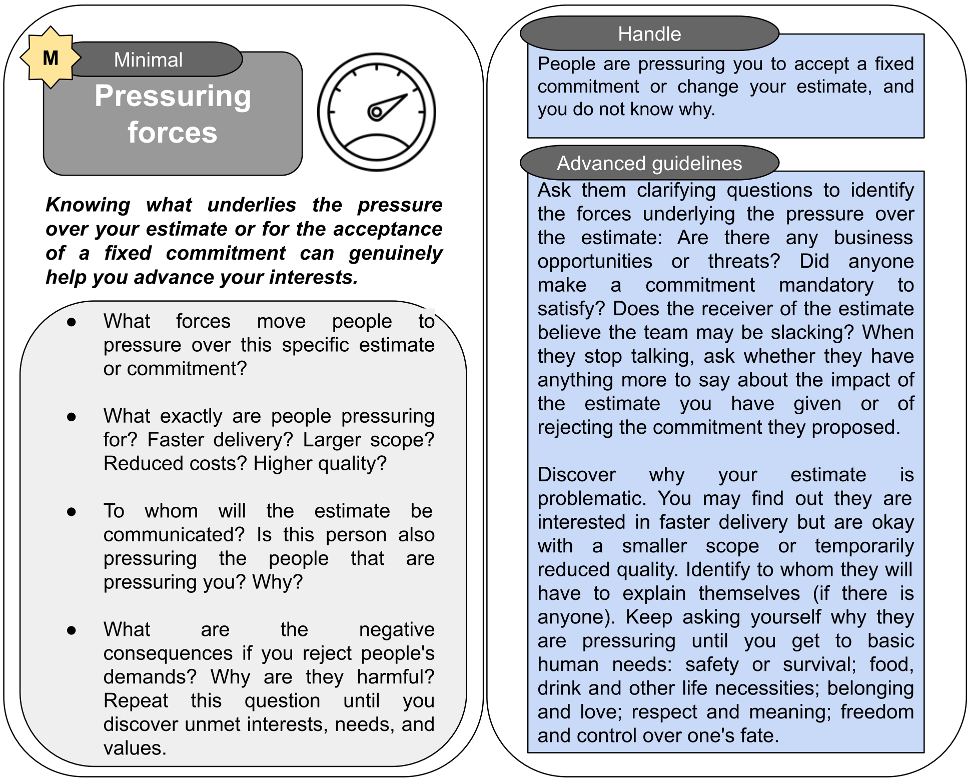


Figure 5.6–- Pressuring forces lens.

This lens is based on the idea of listening attentively to the other side to fully understand their perspective – as part of the Respect your way to Yes Step [20]. Additionally, it involves looking for the interests of the people pressuring the estimator instead of focusing on their position (Focus on interests, not positions), as recommended by Principled Negotiation [18].

The first focusing question aims to investigate more about the pressuring sid''s needs and interests, while the second focusing question aims to understand exactly what the pressure is about. The third focusing question regards a particular type of pressing force: when someone more powerful is pressing the other side. Identifying this can help the estimator put themselves on the side of the ones pressing them by opening eyes to ways to relieve their pressure. Finally, the last focusing question aims to make the estimator get perspective on the negative consequences that can emerge if the other side's interests are not satisfied with unmet basic needs. Table 5.3 presents a set of detailed examples of clarifying questions that can be useful when using the Pressuring forces lens, including justifications and additional follow-up questions.

Table 5.3–- Examples of clarifying questions

“"Is there an internal or external deadline or any other restriction”"	This is only an additional alternative, just in case the other options are unsuccessful in helping to discover what is underlying the pressure over the estimates.	What exactly is the restriction about? In which aspect are we not restricted (costs, quality, scope, deadline)?

The estimator and the software team might not satisfy all the other side's demands but understanding more of their situation can be eye-opening. It is the basis for preparing the ground for commitment proposals that satisfy many of the other side' top-priorities, without harming the estimator interests' (See the Laddering whys and the Candidate commitment lenses for more details on this latter part).

5.5. Laddering Whys Lens

Another relevant lens for the moment of pressure is the **Laddering whys**. The estimator should use it to articulate for themselves and others the interests, needs, and values that are the driving forces for keeping their estimate when facing pressure over a specific estimate or for refusing to accept an imposed commitment. Figure 5.7 shows it.

This lens is about looking for the most profound reasons that lie under the estimate, to the point that the estimator gets to the basic human needs that justify it. It is related to the Uncover your Yes Step [20], grounded on the idea that saying No to pressure is saying Yes to all to these legitimate reasons that estimators need to protect. For this same reason, it also builds on the Focus on interests, not positions principle from the Principled Negotiation [18], making the estimator look for their more intrinsic interests behind their estimate, which is their position.

Table 5.4 presents a set of examples of reasons that estimators can elicit when using the Laddering whys lens. It also identifies why these reasons can be important and their underlying basic needs.

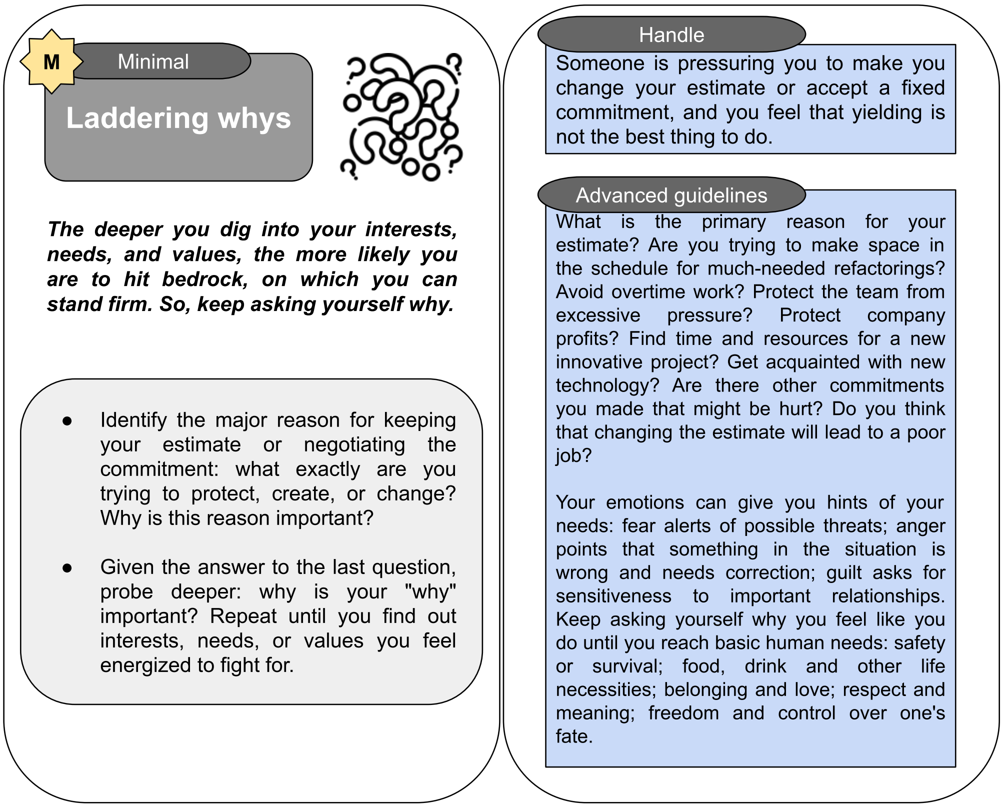


Figure 5.7–- Laddering whys lens.

Table 5.4–- Laddering whys examples.

The reason	Why is this reason important	Why is my“"wh”" important (and underlying basic needs)
I am trying to make space in the schedule for a much-needed refactoring	Because our product quality is deteriorating beyond acceptable levels. This leads us to a high response time to change requests and more errors in production, impacting our customer satisfaction with our service.	Because our client satisfaction is of utmost importance for keeping our business (safety and survival need)
I am trying to avoid overtime work.	Because I have been working overtime often, neglecting time with my family.	Because my family is the top priority in my life (love and belonging need, freedom, and control over on’'s fate need).
There are other commitments you made that might be hurt.	Because we need to keep our word to others, no matter what.	Because we must protect our image of a reliable company/reliable people (respect and meaning need).
I think that changing the estimate will lead to a poor job	Because we need to keep a high-quality standard for our product, and we like to have things done right.	Because we must maintain our image of a quality-focused company (respect and meaning need)

The reason	Why is this reason important	Why is my“"wh”" important (and underlying basic needs)
I am trying to protect the team from excessive pressure	Because excessive pressure cause quality to drop and people to get unsatisfied, leading to high turnover.	Because we have good people that we care about, and it is not easy to find replacements for them (respect and meaning need, safety and survival need)
I am trying to protect company profit margins	Because we need enough money to keep our payroll and cover our expenses. We also want our shareholders happy and the business attractive for them.	Because we need to provide our people a dignified life (food, drink and other life necessities, safety and survival need)
I am trying to find time and resources for a new innovative project	Because we discovered a much attractive business opportunity we must pursue.	Because we want to expand our business (freedom and control over on’'s fate need)
I want to get acquainted with a new technology	Because it can help us to boost our product quality, and I want to keep myself updated	Because I want to be knowledgeable about cutting edge technology and remain attractive for job promotions and opportunities (safety and survival need, freedom and control over on’'s fate need)

In this process, emotions raised during the pressure episode can serve as a signpost for the needs and interests that the estimator wants to protect with their estimate. Fear is an alert of relevant threats; anger suggests something is wrong and needs correction; guilt urges people to be sensitive to their relationships [20].

5.6. Choose your Battles Lens

The **Choose your battles** lens aims to make the estimator rethink whether they really can keep their estimates in a particular situation or whether it is wiser to change it. It also applies to make them reflect whether it is wiser to accept a predefined commitment instead of rejecting it based on their estimates. Figure 5.8 shows it.

This lens aims at making the estimator reassess their decision, based on the Empower your no Step [20]. It recognizes that, sometimes, if the estimator keeps the estimate or rejects the proposed commitment, they will do more harm than good. This lens brings a bit of strategic thinking to the tense moment of enduring pressure. Because it also makes the estimator rethink their true interests, it is also based on the Focus on interests, not positions from Principled Negotiation [18].

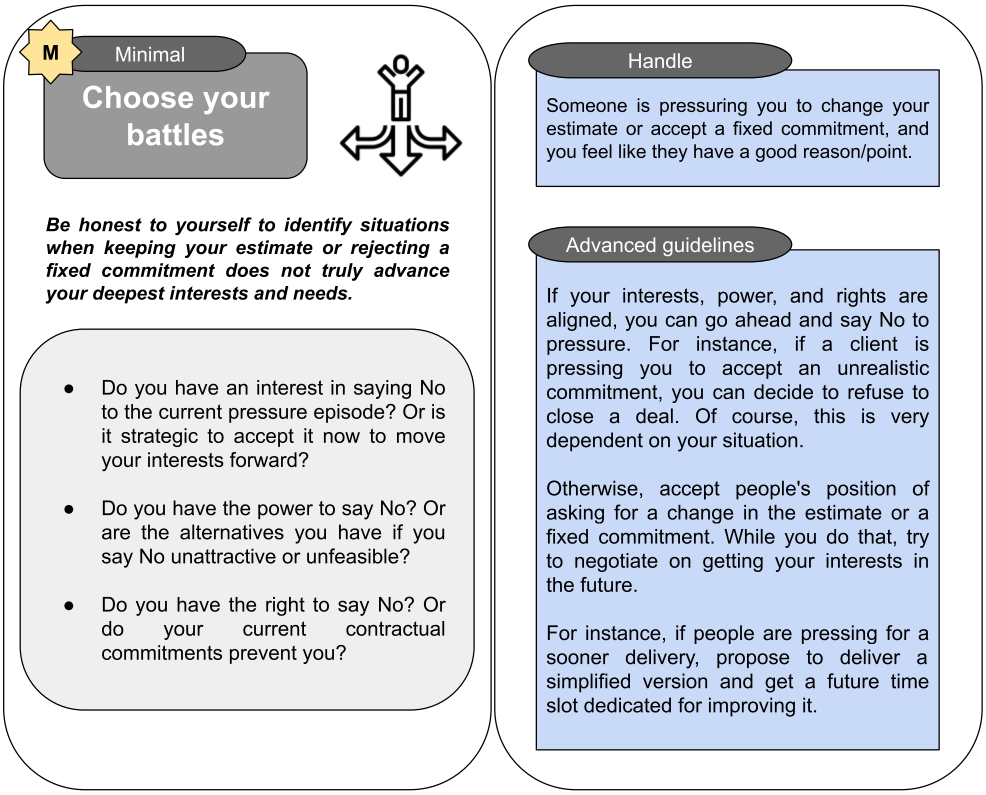


Figure 5.8— Choose your battles lens.

The first focusing question asks the estimator to rethink their interests, especially after understanding the other side’s interests (See the Pressuring forces lens). Estimators might have discovered that one of the other side’s interests is very much aligned with their own in ways they could not predict before. The second focusing question regards power. It requires the estimator to consider the alternatives they have if the other side is not collaborative and keeps pressuring. It also makes the estimator reflect on whether these alternatives are feasible and attractive. The third focusing question requires the estimator to think about whether they are allowed to say No to pressure in the specific situation. Contractual agreements can force them to satisfy a commitment, for instance. Table 5.5 presents a set of examples where estimators have no interest, power, or right to say No to pressure and choose to change their estimate or accept an imposed commitment.

Table 5.5–- Examples of situations where estimators have no interest, power, or rights to say No.

You do not have the…	… because…
Interest	All employees, yourself included, will share the profits of this project, which can severely decline because of fines for late deliveries considering the deadline for the entire project.
Power	This is our only client, and we see no new clients in the next two years prospect.
Right	We already signed a contract restricting the budget, schedule, and project scope.

If the estimator thinks they have the interest, the power, and the right to say No to pressure, the Candidate commitment lens is useful. We discuss it in the following section.

5.7. Candidate Commitments Lens

The **Candidate commitment** lens intends to aid in deriving options of mutually satisfactory commitments. It is about going beyond the estimate to find alternative paths for keeping the estimate while still trying to accommodate the most interests and needs of everyone involved, as Figure 5.9 presents.

This lens is rooted in the Propose a Yes Step [20] and in the principle of Invent options for mutual gains, from the Principled Negotiation [18]. It is about going beyond saying No to pressure to propose a third option: a commitment that aims at satisfying both sides to the greatest extent possible. The two first focusing questions help the estimator derive the most straightforward options as a starting point: the one perfect to them and the one perfect to the other side. The third question is about deriving intermediary options, which are probably more realistic to reach an interesting agreement for both sides. Table 5.6 presents a few options to think of when using the Candidate commitment lens.

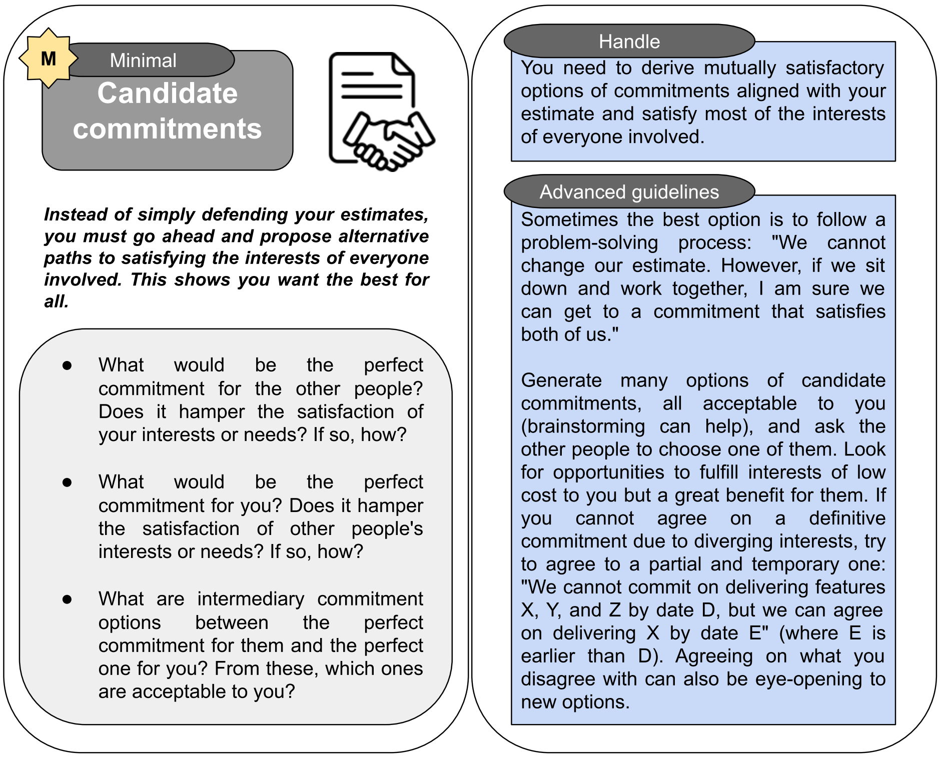


Figure 5.9–- Candidate commitments lens.

Table 5.6–- Commitment options

What are they pressuring for?	Questions to understand further their restrictions	Commitment options
Faster delivery, reduced costs, or higher quality	What are the higher priority features?	- Postpone the delivery of lower priority features
Faster delivery, larger scope, or higher quality	Can we hire additional staff for this project?	- Add more staff to the project, either by hiring new people or by moving people between projects/teams, if it is not too late in the project - Divide the development of the feature with other teams - Keep current staff 100% dedicated to the project
Faster delivery, larger scope, or reduced costs	Can we deliver a simpler version first, and later improve it?	- Simplify the features that will be delivered
Larger scope, reduced costs, or higher quality	Can we be flexible about our schedule?	- Commit to an interval schedule estimate instead of a point estimate - Define revision points in the plan for the schedule project estimate

The Candidate commitment lens is the last one in the Minimal Defense Pack, composing a set of lenses focused on specific pressure episodes. Next, we present the lenses of the Extended Pack, focused on supporting the estimator in the case pressure continues after the first attempts to cease it with the Minimal Pack.

5.8. Keep Strategy Lens

The **Keep strategy** lens is one of the lenses in the Extended Defense Pack. It helps the estimator ground their estimates on legitimate reasons, mostly outside their control, to keep their estimates. That might strengthen their arguments. Figure 5.10 shows this lens.

The focusing questions in the Keep strategy lens present options that estimators can use to ground their No to pressure: policies, other commitments, time-related issues, or even quality issues. They connect the estimate and the commitment to matters that are primarily outside the control of the estimator. This makes the estimate less personal and harder to be rejected. The lens is based on the Assert your No and the Propose a Yes Steps [20], and also on the Insist on using objective criteria from principle from Principled Negotiation [18].

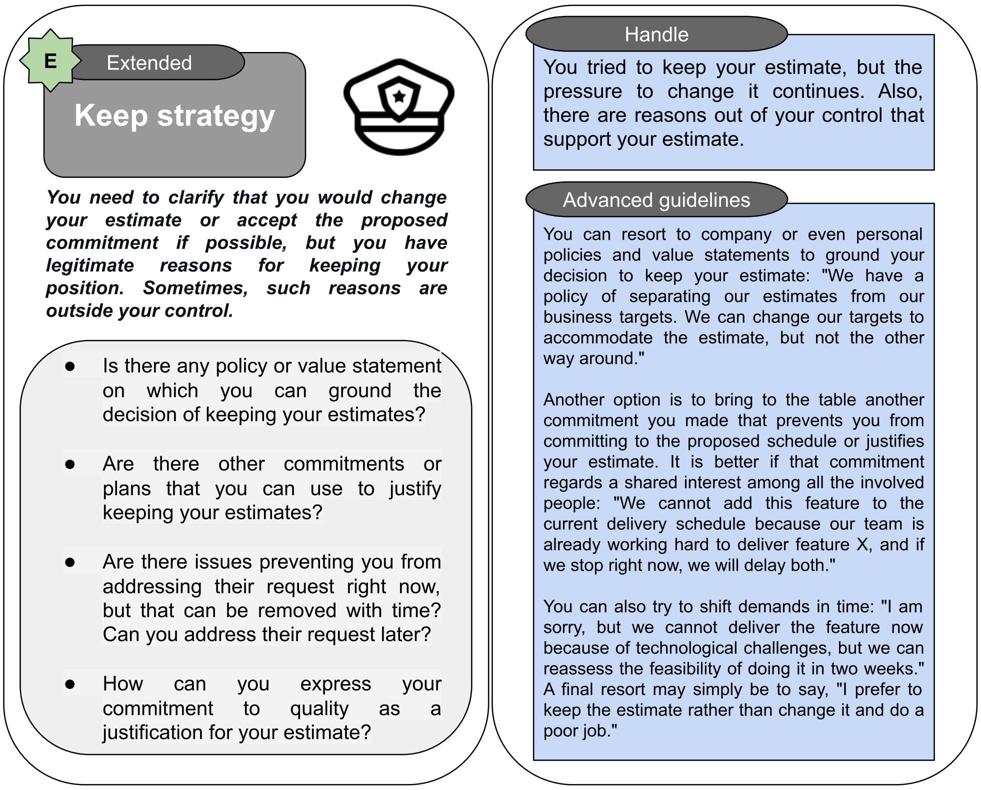


Figure 5.10–- Keep strategy lens.

5.9. Perspective Taking Lens

The **Perspective taking** lens is useful when pressure is getting stronger. It involves making the estimator go to a perspective (like a balcony), where they can free themselves from all the emotions that might impact them negatively, to see the situation more clearly. The other side might be using different tactics like attacks, stone walls, or tricks to make the estimator change their estimate. An attack tries to intimidate and make the estimator uncomfortable; a stone wall is a refusal to budge; a trick will take advantage of the estimato''s beliefs in the other sid''s good faith, deceiving them [19]. Figure 5.11 presents the Perspective taking lens.

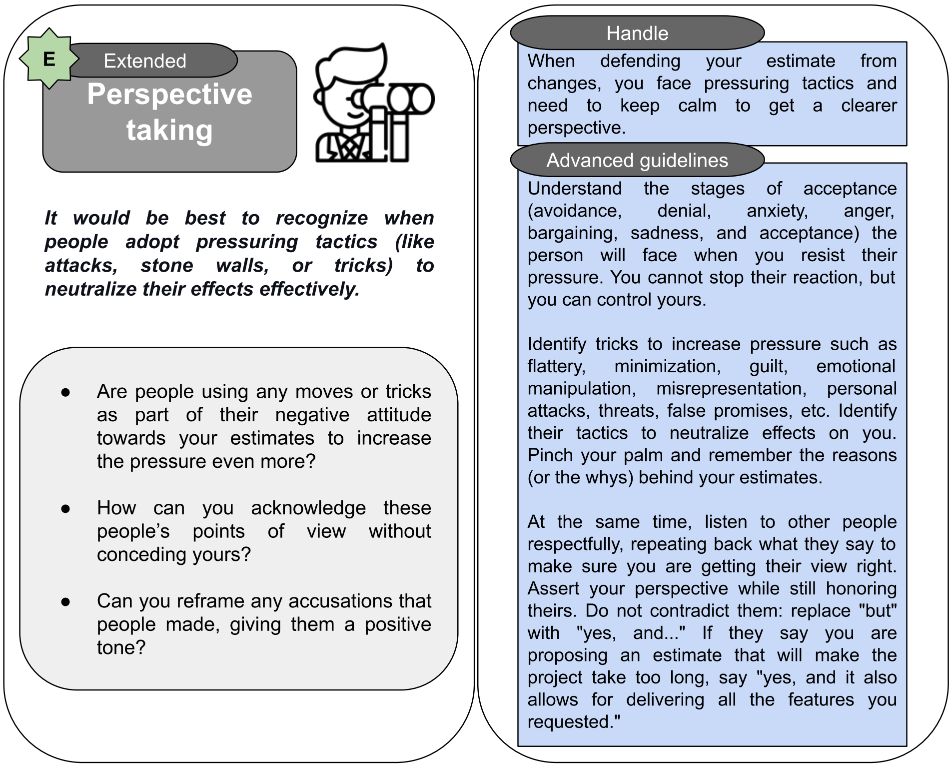


Figure 5.11 – Perspective taking lens.

It is grounded on the Stay true to your Yes Step [20] and the principle to Separate the people from the problem from Principled Negotiation [18]. In addition, it is also based on three steps from the Breakthrough Strategy [19]: (i) Do’'t react: go to the balcony; (ii) Don’t argue: step to their side; and (iii) Don’t reject: reframe. The first focusing question involves identifying their tactics to avoid yielding to the pressure they create. Table 5.7 presents examples of different kinds of these tactics in practice. This lens helps to neutralize their effects and clarify the thoughts to avoid yielding to pressure.

Table 5.7–- Examples of tactics

Tactic	Foundation	Example
Attacks	Based on consequences	“Either you change it or there is no contract!”
	To your proposal	“Your estimates are way out of line!”
	To your credibility	“It looks like you are not so experienced as the rest of your team, uhn…”
	To your authority	“I want to talk with the technical lead, please!”
Stone walls	Previous commitment	“We have already committed with an earlier deadline with the customer. We cannot change that!”
	Final declarations	“It is take it or leave it!”
	Manipulating the data	The other side presents you with a list of features, planning to increase it later on the project.

Tactic	Foundation	Example
Tricks and103therr tactics	Last minute add-on	A last minute new feature is added to the project, right when you thought you had already agreed on the commitment based on the estimates.
	Flattery	“You are the best software team I know! I am sure you can make it to this deadline!”
	Minimization	“But all we need is a small fix on this feature!”

Even if they are attacking or being unfair somehow, the estimator still needs to treat them respectfully, and the second focusing question is about this. The third focusing question is about responding to them to reframe their negativity, reestablishing the path to a mutually satisfactory agreement.

5.10.Reality Test Lens

The **Reality test** lens guides the other side to change their perspective to the natural and logical consequences of changing the estimate and committing to an unrealistic one. By asking the other side reality-testing questions, the estimator can end up showing the point of their estimate. Figure 5.12 presents the Reality test lens.

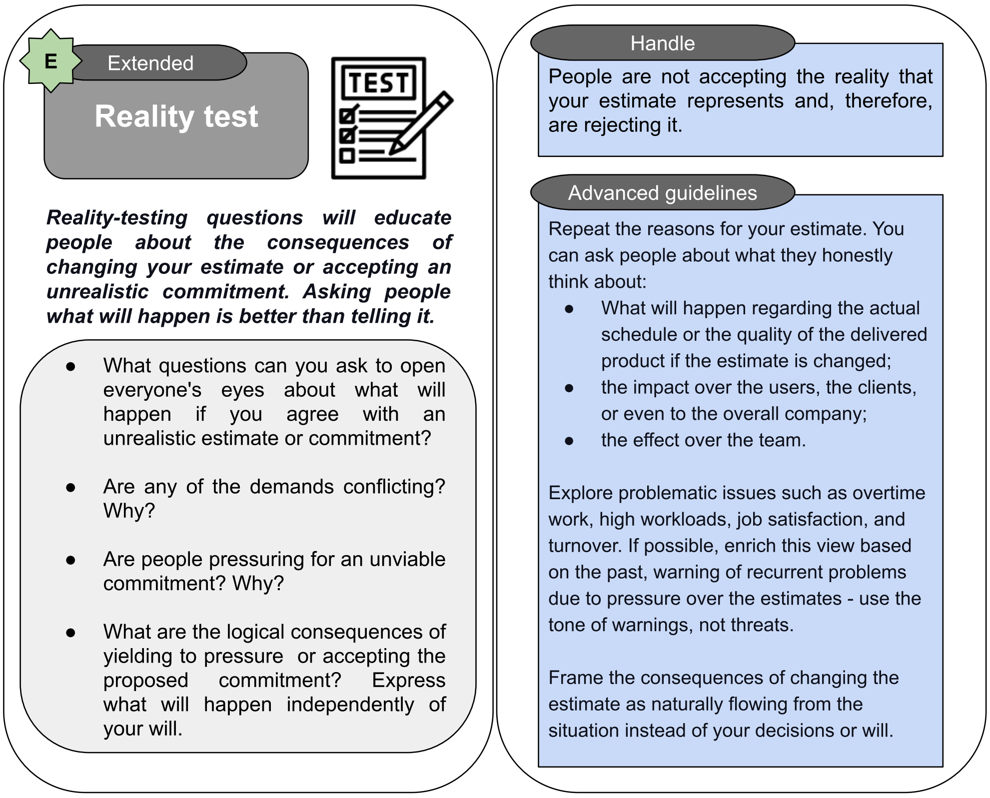


Figure 5.12–- Reality test lens.

The lens is based on the Underscore your No Step [20] and the Do’'t escalate: Use power to educate Step, from the Breakthrough Strategy [19]. The first focusing question regards making them visualize the consequences and reality of yielding to their demands, especially the unjustified and unrealistic ones. Reality-testing questions can aid in this purpose because it allows that they educate themselves, something that can be more powerful than having the estimator telling them the consequences [19]. Table 5.8 provides some examples of reality-testing questions.

Table 5.8–- Examples of reality-testing questions

Impact focus	Example of questions
Schedule-questions	“Ok. Let’s say we commit to the deadline you propose, without any changes to our team, to the list of features we have to deliver and let’s suppose we are keeping our high-quality standards. What do you think will happen if someone in our team gets ill?”
Quality-questions	“All right! Let’s say we commit to the deadline you propose, without any changes to our team or the list of features we have to deliver. We won’t have time to work on the user interface improvements we have discussed before. Do you think the users will still be willing to use the product without these improvements?”
Users, client, or their company-questions	“Right. Let’s say we commit to delivering the product according to your demands. What would happen to your company image if the product failed during your operations because we did not get the time and resources needed to test enough?”
Team-questions	“Fine! Considering we commit to delivering all these features for the next release, how many overtime work hours do you think the team will have to do for the next couple of weeks? How do you think that is going to impact that high turnover issue we have been discussing?”

In any case, if reality-testing questions are not enough, the estimator can also warn them, especially about more technical issues that the other side is unaware of. The last three focusing questions aim at helping the estimator to identify issues to warn about and how to do it. A warning is a prediction about inherent consequences that flow from the situation itself and is different from threatening—which is about imposing consequences yourself [20]. The tone is also different: warnings are respectful and show the willingness to collaborate.

5.11. Golden Bridge Lens

The **Golden bridge** lens helps the estimator build one way so that the other side can retreat from their previous pressure position gracefully. It recognizes they will not accept your estimates if doing it makes them look bad to others. So, the estimator needs to help them to build a bridge from their previous position of pressing for a specific commitment to a new position of accepting a mutually satisfactory agreement. Then, if the estimator believes keeping the estimate is the best, the Candidate commitment lens is helpful. Otherwise, the process ends because all that is left is changing the estimate. Figure 5.13 shows this lens.

The Golden Bridge was designed based on the Negotiate to Yes Step [20] and the Don't push: Build them a golden bridge Step, from the Breakthrough Strategy [19]. The first focusing question asks the estimator to think of other people that the other side might have to respond to and that might not accept the estimate, or the commitment defined based on it. If that is the case, the estimator needs to help the other side to win approval. The second focusing question is about investigating deeper the reasons why they keep the pressure. The other side can still have unmet interests that the estimate or the proposed commitments the estimator made did not address. The third focusing question regards making the other side looking good when accepting the estimate or commitments. This is especially important when the other side has taken a strong position of rejecting the estimate or where the acceptance seems like a loss in a contest of will for them.

This is the last lens in the Extended Defense Pack. Next, we discuss the lenses of the Strategic Defense Pack. This pack is helpful when estimators want to fight against recurrent pressure, improving their working environment.

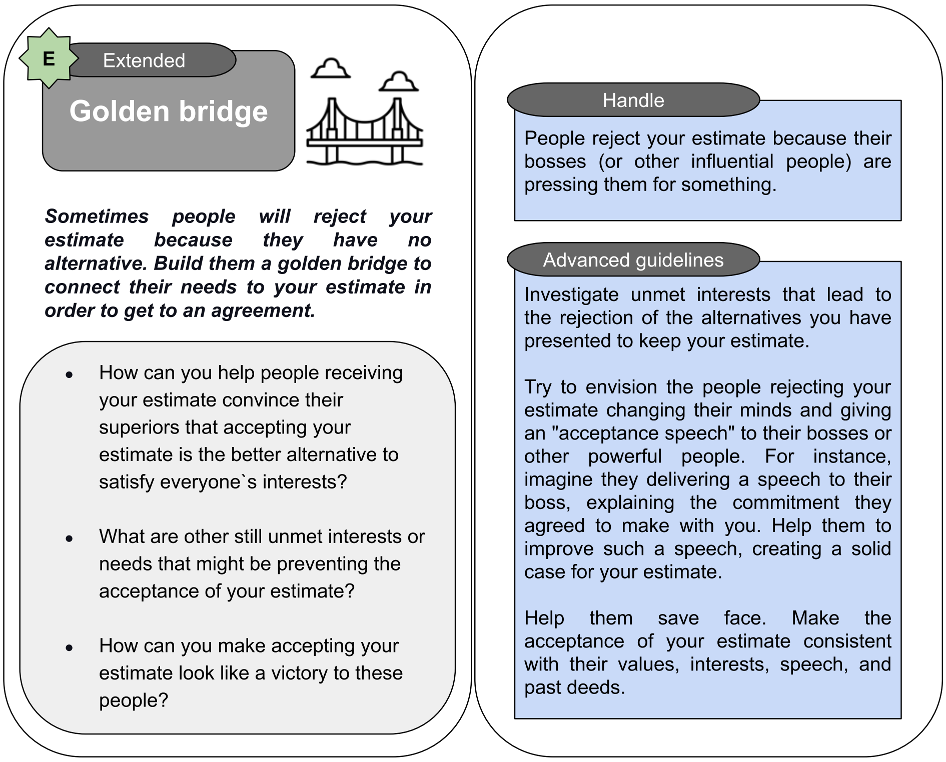


Figure 5.13–- Golden bridge lens.

5.12. Inner Yes Lens

The **Inner Yes** lens is about finding out the most valuable interests, needs, and values that empower the estimator to fight for a better working environment. The estimator can use it together with the Laddering whys lens (from the Minimal Defense Pack) to probe deeper and strengthen the reasons for resisting pressure that leads to unrealistic commitments. Figure 5.14 presents it.

The two first focusing questions require the estimator to elicit their primary reasons for changing their environment, identifying why they feel like they do about past commitments and why they have been accepting demands they should have not. This clarifies the problems they want to eliminate. The last focusing question helps them envision what they want instead of focusing on what they do not want. The lens is motivated by the Uncover your yes Step [20] and the Focus on interests, not positions principle from Principled Negotiation [18].

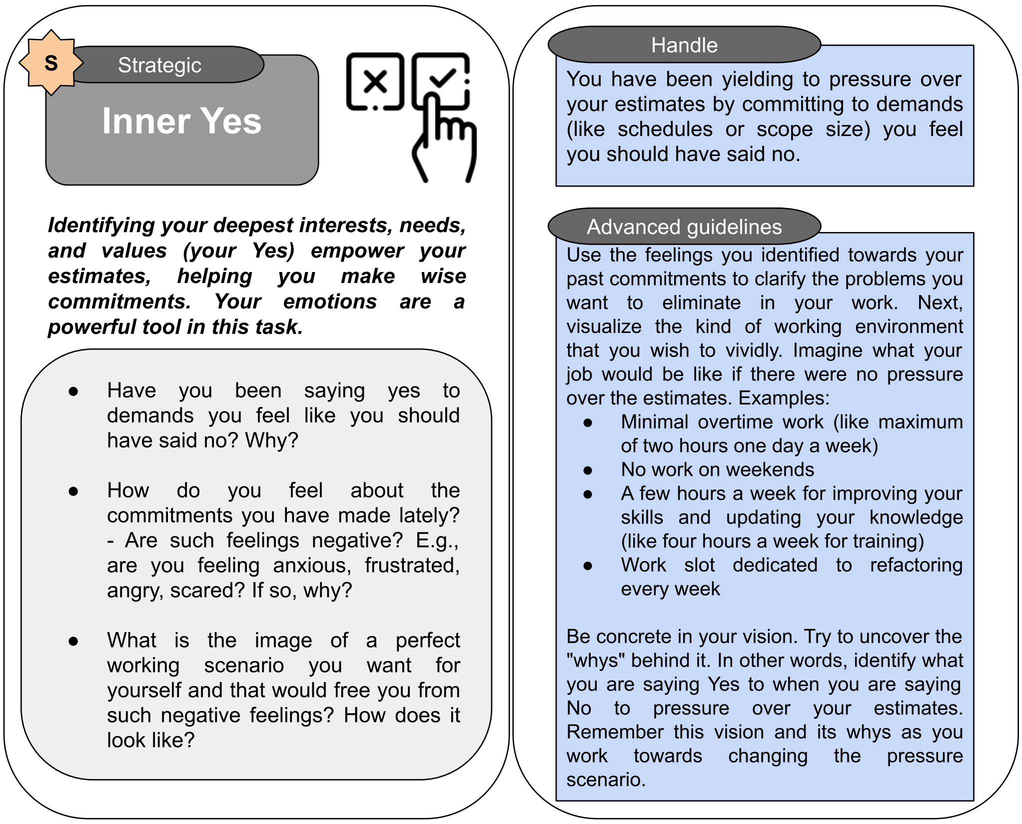


Figure 5.14– Inner Yes lens.

5.13. Reconnaissance Mission Lens

The **Reconnaissance mission** lens is about getting a better comprehension of the other side so that the estimator can think of options for satisfying them while avoiding pressure over the estimates. It relates with the Pressuring forces lens in the Minimum Defense Pack, except that it aims at providing a broader view of the other sides' situation. Figure 5.15 presents it.

This lens is based on listening to the other side attentively, from the Respect your way to Yes Step [20]. The first three focusing questions aim at gaining a better perspective of their situation regarding issues that can be highly relevant for the estimation process and the establishment of commitments, such as whom they respond to, their business goals, and their current market situation. The last question helps the estimator relate the other side's situation to their typical reaction to the estimates. These reactions can hint at interests that the estimates and commitment proposals given in the past failed to meet.

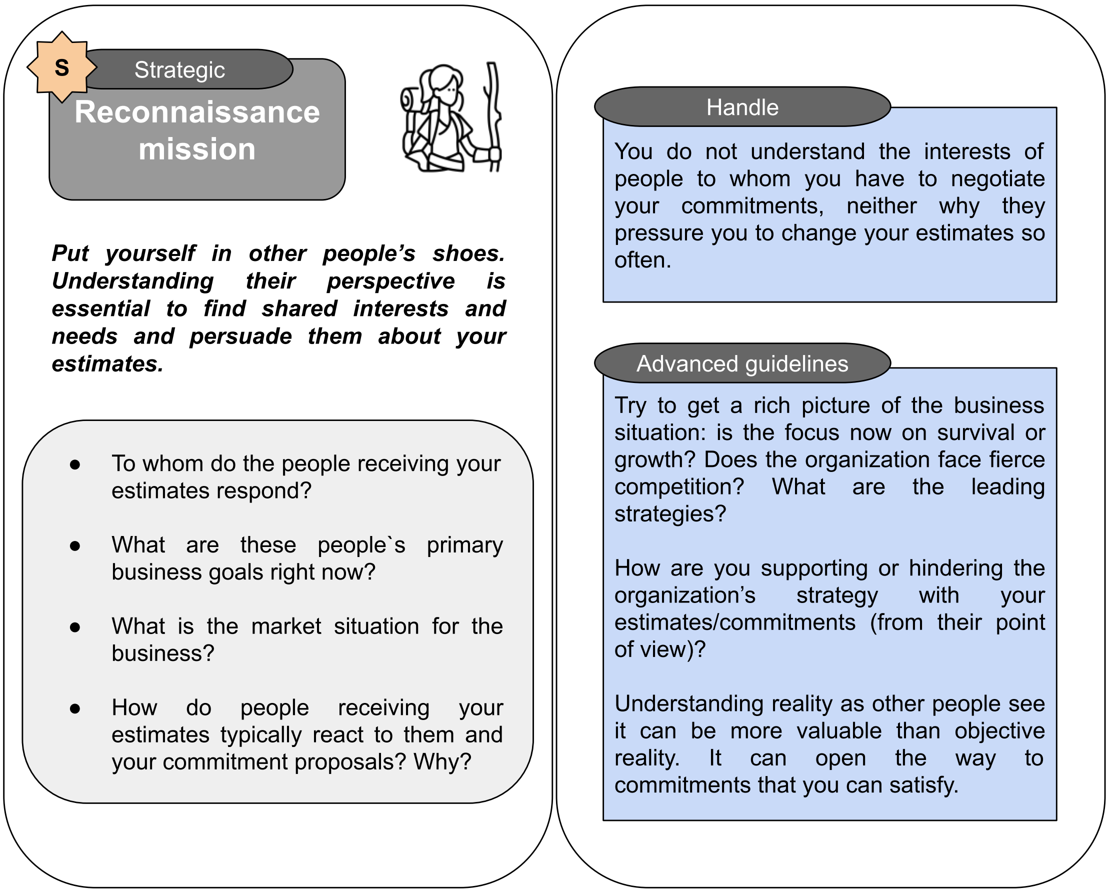


Figure 5.15–- Reconnaissance mission lens.

5.14. Plan B Lens

The **Plan B** lens aims at helping the estimator to find alternatives for when people refuse to change their pressure behavior, even though it is harmful. It is about estimators taking control of their lives and work and defining what they will do independently of others if nothing changes. Plan B is the estimators' BATNA: the best course of action they can take if the other side is unwilling to collaborate [18]. Figure 5.16 depicts the Plan B lens.

The idea of having a Plan B comes from the Empower your No Step [20]. It gives the estimator the psychological freedom to fight for a better environment. The first focusing question helps estimators see the power the other side has on enforcing their demands upon the estimates and commitments. Can they fire the estimator? Can they cancel a vital contract? The second and third focusing questions help the estimator to think of plans for eliminating pressure if the other decides not to collaborate and keep their pressure behavior, thus constructing a Plan B, which can be something as radical as looking for another job. However, a dramatic Plan B can damage the relationship with the pressuring side, so it must be the estimators' last resort, used sparingly.

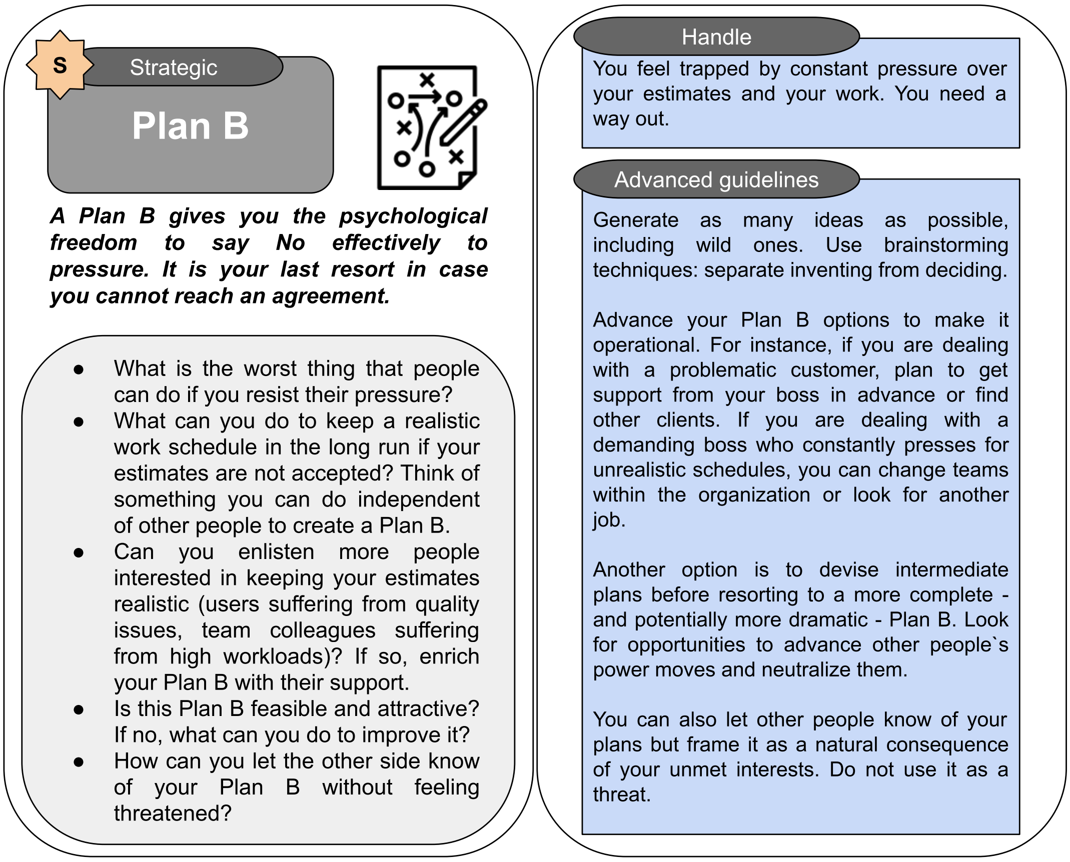


Figure 5.16–- Plan B lens.

The fourth focusing question is about making Plan B more attractive because this empowers the estimator even more. For instance, if they think a good Plan B is to look for another job, they can improve this plan by getting a job offer. Finally, the fifth question is optional and should be applied with care, only if the estimator knows how to make the other side aware of their plans without feeling threatened. The idea is to allow the other side to reconsider their position in light of their acquired knowledge about the estimator power to act independently. This is based on the Don’t escalate: use power to educate [19].

5.15.Let Steam Off Lens

The **Let steam off** lens regards the communication process about how harmful the pressure behaviors are, and that the situation needs a change. It also regards expressing the need for the elimination of pressure for a healthier environment. Figure 5.17 presents it. We framed all the focusing questions to help estimators bring up the fact that there is recurrent pressure and it is not acceptable, based on the Assert your No Step [20]. The lens requires the estimator to invite the other side to stop their pressure behavior, and is also grounded on the Propose a Yes Step [20]. The Let steam off lens is even more potent if the estimator has applied the Plan B lens before. If pressure continues, it is time for the next lens: No bluffing.

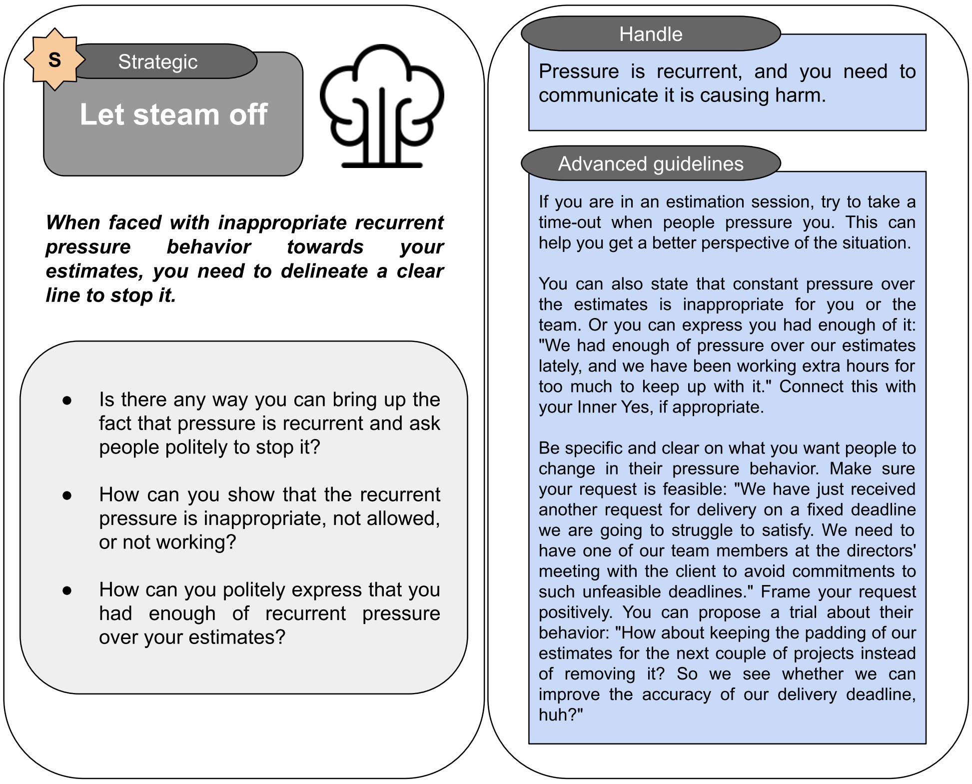


Figure 5.17–- Let steam off lens.

5.16. No Bluffing Lens

The **No bluffing** lens is about executing Plan B most respectfully, keeping the door open for future negotiations. Figure 5.18 shows it.

The lens is based on deploying the Plan B from the Underscore your No Step [20]. The two focusing questions require the estimator to think of ways of doing it respectfully and preserving the relationship with the pressuring side to the greatest extent possible. The estimator can achieve this by clarifying that Plan B is not a punishment but a natural consequence of defending their interests, needs, and values.

5.17. Nourish the Relationship Lens

Finally, the **Nourish the relationship** lens is about looking for the other side's satisfaction beyond formal meetings and work. Figure 5.19 represents it. Instead, it regards building a good working relationship, ensuring them the estimator can be trusted, and avoiding unnecessary pressure over their estimates.

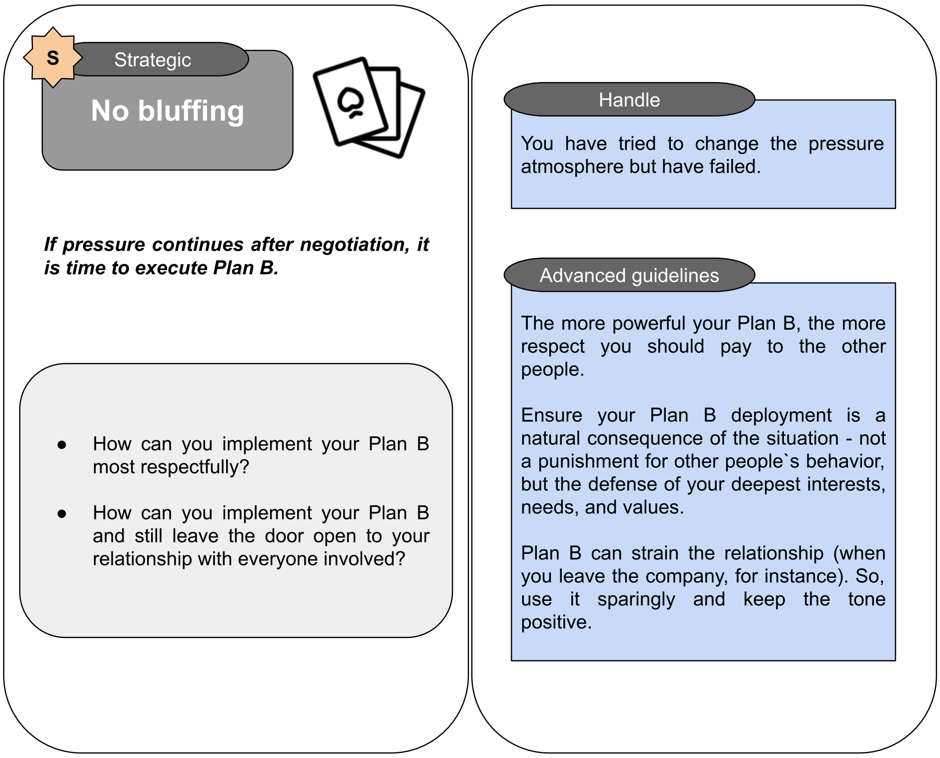


Figure 5.18–- No bluffing lens.

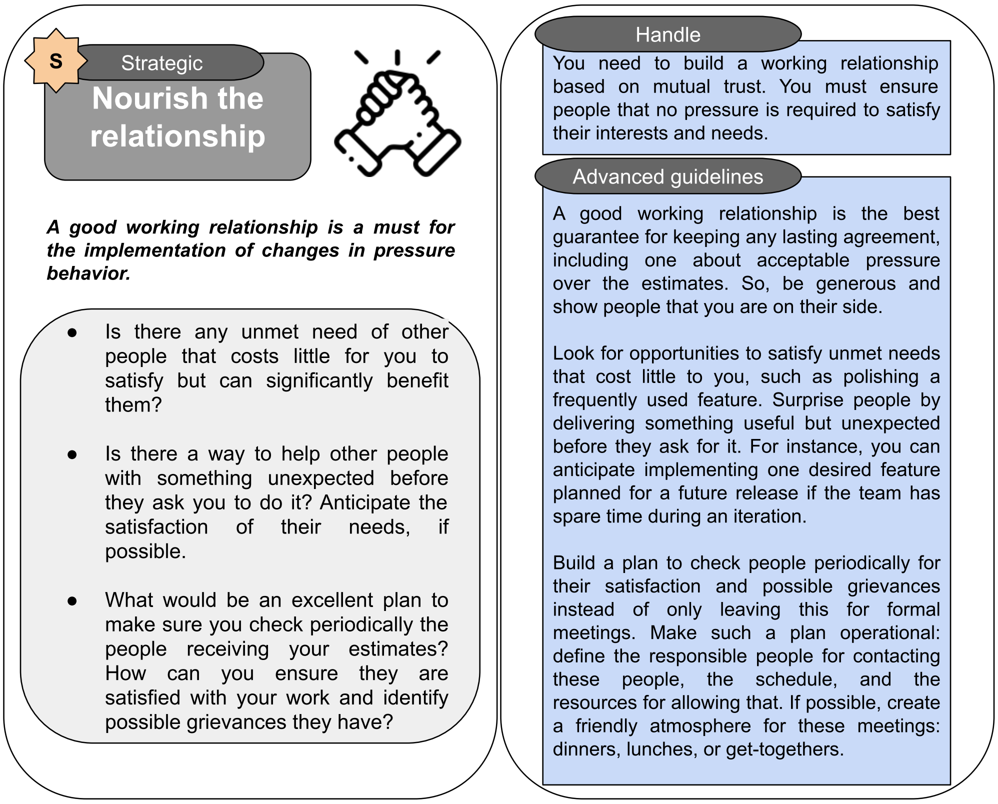


Figure 5.19–- Nourish the relationship lens.

This lens is based on cultivating a healthy relationship, part of the Negotiate to Yes Step [20]. It is also grounded on the idea of forging a lasting agreement, from the Do'’t escalate: use power to educate, of the Breakthrough Strategy [19]. The focusing questions ask the estimator to think of ways to satisfy the other side's needs beyond the estimation and establishment of commitments context. By addressing their needs in a larger context, the estimator shows they are all on the same side, working for the best of all. These lay the foundations for a working environment that can gradually eliminate pressure over software estimates.

5.18. The Digital Simulation

To facilitate learning the lenses and their principles, we designed a digital simulation, which are technology-based simulations that model a process or a system [40]. Simulations provide opportunities to adjust aspects of reality to facilitate learning and practice in varied ways, such as when they address infrequent events or by enabling immediate feedback on the learner'' actions [227]. We implemented it as interactive videos with pressure scenarios and embedded questions, using the platform PlayPosit[19]. We prepared two videos: one for each pack. Figure 5.20 illustrates the video dynamics. Figure 5.21presents a screenshot of the video for this scenario.

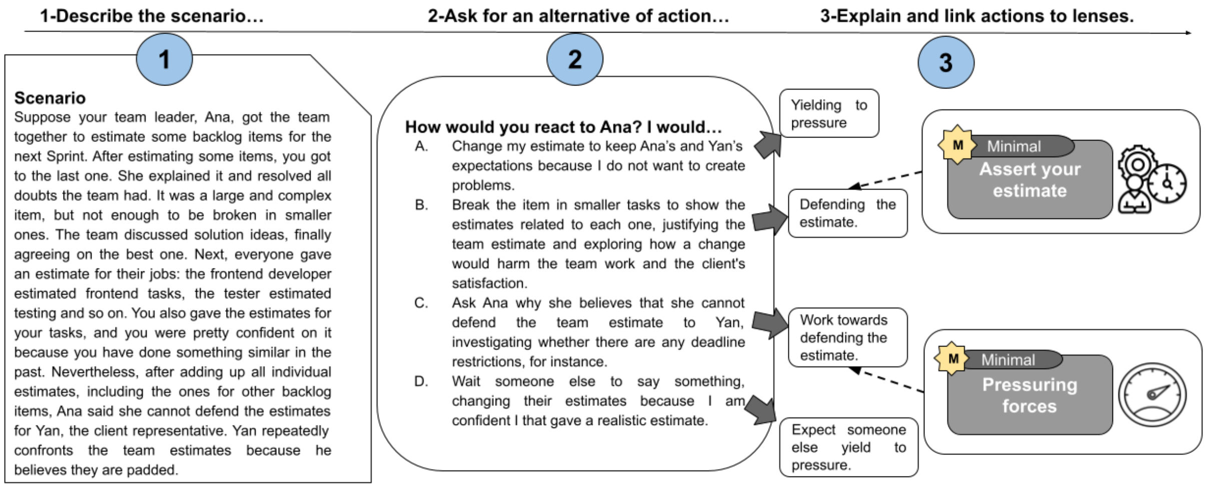


Figure 5.20–- Video dynamics.

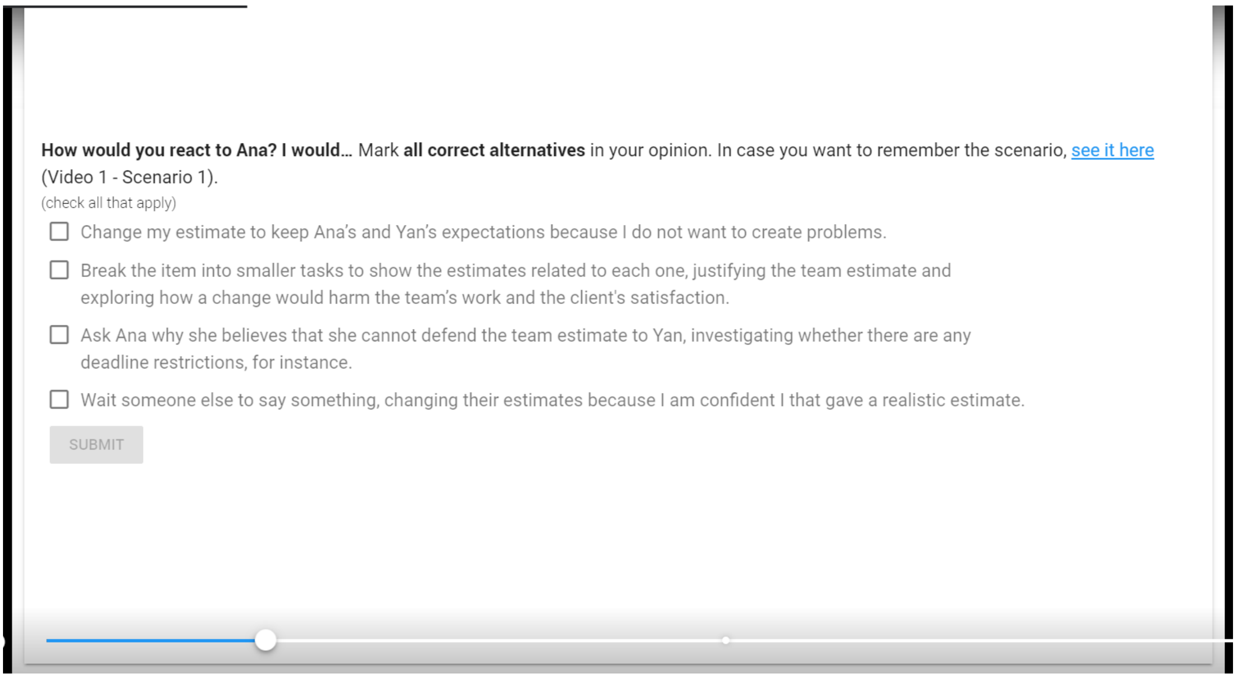


Figure 5.21–- Video screenshot.

[19] https://playpos.it/

After a brief introduction, the video described a realistic written pressure scenario. Figure 5.20.1 shows the first scenario presented in the video about the Minimal Pack. Following, we asked the participant which action they would choose to respond to the scenario of a set of four alternatives. The video paused to allow participants to think. Alternatives included one or two options that represented yielding to pressure—while all others represented paths towards defending the estimates. In Figure 5.20.2, we can see all the options for the specific scenario illustrated in the image. Options A and D represent yielding to pressure or expecting someone else to do it, respectively. After the participant chose the options they wanted, the tool presented a score to indicate whether the answers aligned with the lense'' ideas. For instance, if participants choose options B and C, they get a 100% score. If they choose only one of them, they get a score of 50%. Otherwise, they get a zero score. Next, we discussed each alternative, identifying the ones that represented concessions and connecting the other ones to the defense lenses that supported the depicted behavior. Therefore, as Figure 5.20.3 presents, we identified to participants that options A and B represent yielding to pressure. Then, we explained the lens "Assert your estimate" considering option B and the lens "Pressuring forces" in light of option C.[20]

5.19. Summary

This chapter presented our set of defense lenses, also describing their theoretical foundations. We described the lenses in the format of cards, structured mainly around a handle that indicates when to use the card, the negotiation principle it is built upon, and a set of focusing questions to aid estimators to change their mental perspective from yielding to pressure and changing their estimates to defending them. We also provided software practitioners with advanced guidelines on how to use the lenses and examples, whenever it was possible.

Our set of lenses is composed of an introductory card and fifteen advanced lenses. We organized the advanced lenses in three recommended packs. The first and foremost is the Minimal Defense Pack, with lenses that we designed the to empower estimators to communicate their estimates to avoid pressure and deflect from specific pressure episodes over their estimates or accept unrealistic commitments. The Extended Defense Pack complements the first one, presenting additional lenses that estimators can use in case their first defense attempts fail. The

[20] The first video (in Portuguese) is available at https://app.playpos.it/go/listcode/1507633/1582227/1176427/0/v2---Mdulo-1---Completo
The second video (in Portuguese) is available at https://app.playpos.it/go/listcode/1486380/1582227/1176427/0/v1---Mdulo-2---completo.

Strategic Defense Pack allows estimators to look for opportunities to improve their working environment when it favors pressure over the estimates as the default behavior from managers and clients. Besides, estimators can use each lens in isolation, depending on the situation.

After defining the set of defense lenses as our artifact during our first design cycle, the next step in this research project was to evaluate it, in search of improvement opportunities and to assess whether it effectively contributes to the shift of perspective of estimators, leading them to try to defend their estimates instead of simply yielding to pressure. We conducted a first evaluation through a focus group, detailed in CHAPTER 6. We chose the focus group method because it can help researchers to gain knowledge on personal perceptions of group members on a specific topic [37]. Moreover, the method is low-cost and fast to perform [37], reducing the risks associated with the use of a new technology for the first time. After the focus group, we executed another design cycle, developing the digital simulation to support the gain of knowledge of the lenses. Next, we assessed the digital simulation and its contents through a controlled experiment. In the next two chapters we present the design and the results for the focus group study and the controlled experiment, respectively.

CHAPTER 6 – 1ST EVALUATION: FOCUS GROUP

This chapter presents the focus group that we executed to evaluate the defense lenses. It was a focus group and essentially is part of our design cycle, aiming at the improvement of our artifact.

6.1. Introduction

In the previous chapter, we presented the defense lenses, designed as a result of the information we gained at the rigor and relevance cycles of the DSR approach. This chapter discusses the first empirical study that we executed as part of our design cycle, in which we employed the focus group method [37]. We used the insights we gained to improve the usability and effectiveness of the defense lenses. It also led us to decide on creating the digital simulation to help estimators to understand the lenses.

One critical issue in our evaluation is the scope. Considering that the Strategic Defense Pack aims to help estimators change an environment pervaded by pressure over the estimates and that we have a time limit to any Ph.D. project, we focused on evaluating the Minimal and the Extended Defense Packs.

Section 6.2 describes the materials and procedures that we adopted for the focus group. Section 6.3 describes participants and their background. Section 6.3 presents the results we found.

6.2. Materials and Procedures

We chose the focus group before using other evaluation methods, because it would enable us to improve the lenses at a lower cost/risk for software professionals. The research questions for the study were:

- RQ 1 – Are the defense lenses as described in the booklet useful and usable for defending software estimates against pressure during the establishment of commitments?
- RQ 2 – What are the improvements to the defense lenses as described in the booklet?

We selected participants from our network, focusing on covering different perspectives in terms of experience and roles. Thus, we invited people with varied experience with software engineering in terms of years of experience (ranging from one to 23 years) and roles (including people with experience as software developers, testers, and managers).

After selecting participants, we asked them to participate in a role-playing focus group in two different meetings. The first meeting is based on the following steps:

1. Gather all participants in a one-hour meeting (maximum) to explain the study objectives and get their informed consent.
2. Present the "Pressuring forces" and the "Laddering whys" lenses in detail as a starting point to acquaint them with the lenses. Describe the structure of the booklet containing the whole lenses set and give them a copy for further study.
3. Collect data about participants' personality traits (Big-Five Inventory, with 20-items [228]) and assertiveness levels (Rathus Assertiveness Scale [229], Brazilian version with 28 items), because these can be intervening variables in our study.

The second meeting happened one week after the first. We executed the following procedures:

1. Gathered all participants to resolve any questions they have about the lenses. At this point, collected the data about their doubts.
2. Asked someone to volunteer to represent the side receiving the estimate (a client or a manager) from now on referred to as the receiver. All the other participants played the role of estimator.
3. We randomly chose one scenario from a pool. This scenario represents a real-world situation related to the estimation and the establishment of commitments, where the estimator is expected to apply the lenses. The estimators remained unaware of the chosen scenario. The receiver is expected to conduct the task in the next step according to it.
4. While the receptor studied the chosen scenario and the software specification, the estimators read the specification and defined their estimate. The specification was about a software project entitled SeminarWeb [156], which we chose after searching for detailed specifications used in previous studies about software effort estimation.
5. Considering the chosen scenario and the defense lenses, the estimators and the receiver worked together to agree on a commitment. The estimators presented the estimate, and the receiver applied pressure at the appropriate point, as described in the scenario.

6. After they reached a conclusion, we collected data about the expected outcome from the point of view of the estimators and the receiver. We also asked about the lenses that the estimators used.
7. Executed a final debriefing session to gather overall impressions from the participants regarding the lenses and their improvement suggestions.

We present the scenarios that we created in APPENDIX D. We also included their translation to Brazilian Portuguese in APPENDIX E. For the selected scenario, we expected participants to use the lenses from the Minimal Pack, especially the Assert your Estimate, Pressuring Forces, and Candidate Commitments lenses. The meetings were all remote, so we created and managed videoconference rooms for them as needed.

6.2.1. Data Collection and Analysis

Right after the second meeting, participants answered a questionnaire to collect demographic data. Moreover, an intervening variable in our study is **personality** because previous literature has shown a connection between personality and conflict resolution styles [230]. For instance, we can expect that people who score lower in openness to experience can be more resistant to accepting to use the defense lenses. Therefore, we collected data on the first questionnaire using the Big-Five Inventory (BFI), with 20-items (BFI-20) [228], which assess individuals regarding five personality traits: extraversion, agreeableness, openness to experience, conscientiousness, and neuroticism. We chose BFI-20 because it is short, a characteristic that makes it more appropriate for industry studies, where participants have little time available. An additional advantageous feature of BFI-20 is that it has been validated for the Brazilian population. We included the BFI-20 in ANNEX A.

Another intervening variable is the individual assertiveness level of the estimator. Interpersonal assertiveness is the degree to which people speak out and stand for their interests when those interests are not perfectly aligned with others' [53]. When an estimator respectfully defends their estimate in the face of pressure, they exhibit assertive behavior. When they yield to pressure and change their estimates even when they know this will lead to unrealistic commitments, their behavior is unassertive or passive. Therefore, we considered it important to know more of the general assertiveness characteristics of participants in our study and measure it using a Brazilian adaptation of the Rathus Assertiveness Scale (RAS) [229], composed of 30 items, as we show in ANNEX B.

During the second meeting, data collection happened at three points. The first regards doubts about the lenses, at the meeting onset. We analyzed these data to identify clarifications needed in our artifact. The second point was right after the execution of the simulation with the scenarios. We asked participants to answer each of these questions individually:

- What is your expected result for the scenario regarding whether the estimate is changed or defended?
- Which lenses did you apply? (for estimators only)
- Do you think the lenses contributed to the scenario result? Why (not)?
- Were there specific lenses you did not understand how to apply? Which ones? (for estimators only)
- Were there specific lenses you chose not to apply? Why? (for estimators only)

Next, the third point of data collection was a debriefing session guided by the following questions:

- What do you think we need to change/improve in the lenses or the booklet to make them usable in real scenarios that you face (or faced) when estimating and establishing a commitment?
- Do you have any suggestions to make the lenses more useful for the actual estimating and commitment establishment scenarios you face? Do you have any suggestions to make the lenses more usable (easy to consult and apply) for the actual estimating and commitment establishment scenarios you face?
- Do you have interesting examples (like the ones in the booklet) from your professional lives that can contribute to the lenses or the booklet? If yes, can you share them with us?
- Is there anything else you would like to mention/discuss?

Table 6.1 summarizes all the data we collected and their relationship with the second meeting steps and our focus group' research questions.

Table 6.1–- Summary of data collection

Step	Data	RQ
1	Questions raised by participants regarding the lenses	2
5	Issues about the uses of the lenses	1

Step	Data	RQ
6	The expected result for the scenario (estimate is changed or defended)	1
6	Reasoning for the expected result	1
6	Lenses applied by the estimator	1
6	Reasoning regarding whether the lenses contributed to the expected result	1
6	Lenses that the estimators did not understand	2
6	Lenses that the estimator chose not to apply	2
7	Overall impression about the negotiation lenses	2
7	Overall improvement suggestions for the lenses	2

The data analysis focused on coding the answers and observations of participants to identify the improvements that we can make to our artifact, using the Grounded Theory procedures [220]. We implemented these improvements as changes to the lenses and to the supplemental material (the examples).

6.3. Results

We had five participants from the software industry, represented by a participants' ID in Table 6.2. It shows the participants' experience in the software industry (in years), their Big-Five personality traits' scores, and assertiveness level score. The highest scores are highlighted.

Table 6.2–- Focus group participant'' profiles.

pId	Experience	Openness	Conscientiousness	Extraversion	Agreeableness	Neuroticism	Assertiveness
PFG1	**23**	15	17	11	16	**15**	43
PFG2	1	14	14	15	14	12	51
PFG3	1	6	11	8	14	**17**	42
PFG4	12	**17**	**18**	**19**	**18**	11	**57**
PFG5	**23**	**20**	**19**	**17**	**20**	13	**69**

We selected participants from our network, focusing on covering different perspectives in terms of experience and roles. Thus, we invited people with varied experience with software engineering in terms of years of experience (ranging from one to 23 years) and roles (including people with experience as software developers, testers, and managers).

The pressure scenario randomly selected for the group dynamics was "The Trade Show" (Scenario 4 in APPENDIX D). Table 6.3 presents the lenses each participant said they used (U), they did not understand (Nu), or they decided not to apply (Na). The cells with an hyphen represent lenses that participants did not mention in the questionnaire.

Table 6.3–- Participants' uses of the lenses.

PId	Wild	Ladd	Pres	Choo	Asser	Cand	Keep	Pers	Real	Gold
PFG1	Nu	U	U	U	U	U	Na	-	-	Nu
PFG2	Nu	-	Nu	-	U	U	-	-	-	-
PFG3	-	-	-	U	U	U	-	-	-	-
PFG4	Nu	U	Na	Na	U	U	Na	-	Na	-

Table 6.3 shows that participants collectively used all lenses from the Minimal Pack, especially the **Assert your estimate** and the **Candidate commitments**, as we expected. This indicates that they perceived the set of lenses as useful given the scenario at hand. Unexpectedly, only PFG1 chose to use the **Pressuring Forces** lens, and PFG4 thought it would be better not to apply it at all (Na in Table 6.3). Also, P2 did not understand it (Nu in Table 6.3), revealing a need to improve this lens. Also, as we expected, participants decided not to use the lenses from the Extended Pack. PFG4 informed explicitly that he did not apply some of the lenses of this pack, because he believed they would be useful only for inflexible scenarios when more steady guidance is needed—showing that he understood the purpose of this pack well.

Three participants indicated they did not understand the **Wildcard**. PFG1 explained that she had difficulty getting the idea because the wildcard metaphor was broken: when learning a game, one usually learn all its rules and details. The wildcard is presented at the end as a versatile card. However, it was the first card presented in our booklet and served as a guide for some of the other ones. Participants also indicated other improvement opportunities for the presentation format: the current booklet format is helpful for training purposes, but it lacks an indexing system to quickly search for the right lens when the situation requires the estimators to respond fast. We designed the handle and the Wildcard to provide such aid, but these elements were not entirely effective. These preliminary results showed that participants perceived the defense lenses as useful and to a large extent understandable, although improvements were still needed.

Another interesting result comes from PFG3—the estimator with low assertiveness, and one of the least experienced participants. He reported using few cards, possibly explained by the low score in the openness trait. In addition, he discussed during the debriefing that he does not feel comfortable arguing with other people, especially more experienced ones. He believes that example sentences that the estimator can pick from the lens and use directly could help overcome this. Such issues reveals that more inexperienced software practitioners can benefit from more support for learning how to use the lenses. Interestingly, PFG4—the estimator with a high assertiveness profile—made a counterpoint that the set of lenses as a whole helps the estimator to gather arguments together. This shows that even people who already have an appropriate assertive behavior can benefit from the lenses in the estimation context.

6.3.1. Summary

In the current chapter we present the first empirical study that we carried out as part of our design cycle. We had five software professionals participating of a focus group, where four of them acted as estimators while one of them acted as someone receiving estimates and putting pressure over estimators. Estimators collectively used the cards from the Minimal Pack, as we expected for the scenario. Although they understood most of the lens, the focus group also revealed improvement opportunities for our lenses and their presentation format. Therefore, we redesigned our artifact to include a digital simulation, thus supporting more inexperienced participants while also providing a more dynamic format to presenting the lenses—as we showed in Section 5.18. We also improved the writing of the lenses, included more examples on the booklet, and excluded the Wildcard from the last version of the lenses' description. Next, we discuss the controlled experiment that we carried out to evaluate the digital simulation together with the defense lenses.

CHAPTER 7 – 2ND EVALUATION: CONTROLLED EXPERIMENT

This chapter presents a controlled experiment that we conducted to assess the digital simulation and the defense lenses to promote the alternative of defending software estimates instead of yielding to pressure.

7.1. Introduction

After improving the defense lenses and designing the digital simulation to present them, we executed a controlled experiment with industry practitioners, which we describe in this chapter. As we mentioned in CHAPTER 1, we were interested in understanding how the digital simulation and the defense lenses can impact software practitioners' behaviors in their daily practices—a concrete step toward Behavioral Software Engineering [38]. Thus, we examine whether participation in the digital simulation affects professionals' intentions to adopt the strategy we proposed, thus moving on from the software engineers' gambit. We considered that intentions are the immediate antecedent of behavior, as posited by the Theory of Planned Behavior (TPB) [41]. We also collected data on attitudes, subjective norms, and perceived behavioral control, as these are antecedents of intentions. By analyzing data on TPB, we take advantage of an existing social science theory as part of the foundations of our work, something still surprisingly uncommon in Software Engineering research [55].

In Section 7.2, we present our experimental design, including research questions, hypotheses, and data collection and analysis procedures. In Section 7.3, we present our results. In Section 7.4, we present the discussion, including issues of practical relevance for industry. In Section 7.5 we discuss the threats to validity to this study. In Section 7.6, we provide a summary.

7.2. Experimental Design

In this study, we addressed the following research questions:

- RQ 1 - Does the participation in the digital simulation increases software practitioners' intentions to defend their software estimates, as well as its antecedents (attitudes, subjective norms, and perceived behavioral control)? And
- RQ 2 - What is the perceived usefulness of the defense lenses in real-world situations from the perspective of participants of the digital simulation?

To answer them, we carried out a controlled experiment including an experimental and a control group. We exposed the experimental group to the digital simulation and to the defense lenses as described in CHAPTER 5, expecting it to stimulate changes in the intentions of defending software estimates. For comparison, we exposed a control group to reflection questions about past pressure scenarios they faced in their jobs and the impact of such pressure. If we can get higher intentions to defend software estimates from participants in the control group, we would have a much simpler intervention to propose for estimators. Thinking of pressure scenarios and their consequences immediately before communicating estimates would be easier and cheaper than studying the defense lenses through the digital simulation.

We collected data in pre- and post-questionnaires based on the Theory of Planned Behavior and on the reaction of participants to a set of pressure scenarios, as described in more detail in Section 7.2.2. We piloted the questionnaires with four participants: two in the control and two in the experimental group. We improved the wording of a few items. During the pilot study, we also included the reflection questions in the post-questionnaire of the experimental group—a decision we changed for the final study. Moreover, we did not include the pilot participants' data in our final analysis.

Therefore, considering the previous literature suggesting that we can use negotiation principles in defense of software estimates, we hypothesized that after participating in the digital simulation and learning about the defense lenses, participants in the experimental group would exhibit:

- …**higher levels of attitude** to defend their software estimates than before participating (H1a) and than participants in the control group (H1b).
- ... **higher levels of subjective norm** to defend their software estimates than before participating (H2a) and than participants in the control group (H2b).
- ...**higher levels of perceived behavioral control** to defend their software estimates than before participating (H3a) and than participants in the control group (H3b).
- ...**higher levels of intention** to defend their software estimates than before participating (H4a) and than participants in the control group (H4b).

In addition, we assessed the participants' reaction to pressure scenarios we designed inspired by the SE literature, believing participants in the experimental group were more likely to pick alternatives representing the defense of estimates. Thus, we also hypothesized that:

- After participating in the digital simulation and learning about the defense lenses, participants in the experimental group **will choose more defense actions** in pressure scenarios than participants in the control group (H5a).

For each of these research hypotheses, we have a corresponding null hypothesis, stating that after participating in the digital simulation, participants in the experimental group would **not exhibit higher levels** of each given variable than before participating (for H1a-H4a) and than participants in the control group (for H1b-H4b). For instance, the corresponding null hypothesis for H1a states that participants in the experimental group **do not exhibit higher levels** of attitudes to defend their software estimates than before participating. Also, the corresponding null hypothesis for H5a states that after participating in the digital simulation and learning about the defense lenses, participants in the experimental group **will not choose more defense actions** defense actions in pressure scenarios than participants in the control group. Next, we discuss our sampling strategy and provide information about the study' participants.

7.2.1. Sampling Strategy and Participants

We invited 75 people with varying experience in software development and maintenance, both in terms of roles and years of work, from our network to participate in the experiment. We made individual contact with each candidate to participate through LinkedIn, asking whether they were working with software development and whether they were involved with software estimation. If answers were positive for these two questions, we asked them about the estimation process of their teams and companies, to identify whether they communicated their estimates and made commitments based on them. A total of 45 people accepted to participate in the study. We randomly assigned 23 to the control group and 22 to the experimental group. From these, 32 participated in all the study stages: 16 in the control group and 16 in the experimental group. In Table 7.1 we present the descriptive statistics for demographics of participants in each group.

Table 7.1 - Participants' demographics.

	Control n = 16	**Experimental** n = 16
Gender	Men = 11	Men = 13
	Women = 4	Women = 3
	Other = 1	Other = 0
Age	Mean = 30.6 (sd = 5.6)	Mean = 30.1 (sd = 5.0)
Experience	Mean = 8.1 (sd = 4.9)	Mean = 7.3 (sd = 4.8)
Education level	High School or Bachelor’s/College Degree = 13	High School or Bachelor’s/College Degree = 10
	Masters or PhD Degree = 3	Masters or PhD Degree = 6

The control group participants had almost one year more experience in software development and maintenance than the participants in the experimental group on average. Visual inspection of the data reveals the control group is slightly more experienced—but the difference was not statistically significant. As for educational level, the experimental group had a few more participants with a with at least a master’s degree than the control group. We also collected data from participants’ roles. Around 90% of participants in the experimental group and 80% in the control group reported that they work as Developers, Machine Learning Engineers/Data Scientists, Tech Leads, or Agile Experts. The rest of the participants reported being managers, product owners, or requirements engineers.

7.2.2. Data Collection Procedures

To collect data on the intention to defend software estimates and the other TPB variables, we built a questionnaire following the instructions in Francis et al. [61]. We derived 15 questions: three for intentions and four for each of intentions’ antecedents (attitudes, subjective norms, and perceived behavioral control). All items follow a seven-point Likert scale with neutral. For instance, one of the items regarding perceived behavioral control was: “I am confident that I could defend an estimate when facing unreasonable pressure if I wanted to”, with options going from “strongly disagree” to “strongly agree”.

To assess the reaction of software practitioners, we also derived scenarios representing situations with pressure over software estimates. In each scenario, the participant had to choose one of four alternative action options—where one always represents the behavior of defending the software estimates. In contrast, the others represent the behavior of yielding to pressure. The complete list of questions and scenarios is part of APPENDIX F.

Both groups answered the first questionnaire in the first moment (M1), with demographic and TPB questions. We sent the questionnaire via e-mail to all participants, asking them to answer it within one week. In the second moment (M2), the operationalization of the study changed for each group.

Participants in the experimental group engaged with the digital simulation and were exposed to the defense lenses. We sent them the links to the interactive videos, giving them one week to watch. At the end of the video, we left a link for the final questionnaire, with the same questions regarding TPB from M1 plus questions about the actions they would take in the five pressure scenarios. We also asked whether they considered the lenses and negotiation principles would help deal with pressure in their work. If they answered "yes", we also asked which lenses or principles they considered the most useful and why.

In M2, participants in the control group answered questions about pressure scenarios they faced in their jobs and what was their typical outcomes. Next, we asked them to answer the TPB and scenario questions. By doing so, we expected to create a priming effect regarding past pressure experiences from participants. Primes are used in psychology research to selectively increase the accessibility of specific conceptions or pieces of information in memory [231], leading to changes in behavior. For instance, past research has shown that we can prime power in applicants for jobs, making them feel either powerful or powerless immediately before the writing of application letters or interviews, and either improving or worsening their application outcomes, respectively [232]. Therefore, we expected the past pressure scenarios and their outcomes to prime the typical behavior of participants in such situations, creating a baseline behavior for comparison.

7.2.3. Data Analysis Procedures

In our data analysis, we employed the traditional Null Hypothesis Significance Testing (NHST). However, the Software Engineering research community gets increasingly aware of its limitations—such as the rejection of the null hypothesis based on the probability $\mathbb{P}[\text{data}|H_0]$ (p-values), when we need the posterior probability $\mathbb{P}[H_0|\text{data}]$ to accept or reject a hypothesis based on empirical data [233]. Moreover, with NHST there is a higher chance of rejecting the null hypothesis as the number of observations grows because it is usually more restricted than the alternative hypothesis. So it gets likelier that some effect is detectable [234]. The research community has proposed using Bayesian Hypothesis Testing as an alternative [235]. Thus, we also adopted it to compare the plausibility of the research hypotheses and null ones *relative to one another*.

We tested whether M2varE > M1varE, i.e., we assessed whether the variable of interest (attitude, subjective norm, perceived behavioral control, or intentions) was higher for the *experimental group* at Moment 2 (post-questionnaire) compared with Moment 1 (pre-questionnaire). We used the Wilcoxon Signed-Rank paired samples test for that. We also assessed whether M2varE > M2varC, i.e., we tested whether the variable of interest at Moment 2 was higher for the *experimental group* than for the *control group*. In this case, we used the Mann-Whitney test. In all cases, we applied one-sided testing because previous literature supports the idea that negotiation principles could aid in defending software estimates—a reason to believe that results would increase in the experimental group after the intervention and when compared with the control group.

Furthermore, we carried out a reliability analysis for the TPB questionnaire. We dropped one item for attitudes, one for subjective norms, and one for perceived control to improve Cronbach's α score. We give more details on this in Section 7.5 and APPENDIX G.

7.3. Results

So, does the participation in the digital simulation and learning about the defense lenses increases software practitioners' intentions to defend their software estimates, as well as its antecedents? We focus on answering this first question in Section 7.3.1. Also, what is the perceived usefulness of the defense lenses in real-world situations from the perspective of participants of the digital simulation? We focus on this second question in Section 7.3.2.

7.3.1. RQ1: Intention, Its Antecedents, and Pressure Scenarios

As Section 7.2 describes, we investigated the impact of the digital simulation exposing the defense lenses compared with reflecting on pressure scenarios over the intentions of estimators to defend their software estimates. In Figure 7.1, we present the boxplots of the TPB variables of the experimental and control group before and after the intervention.

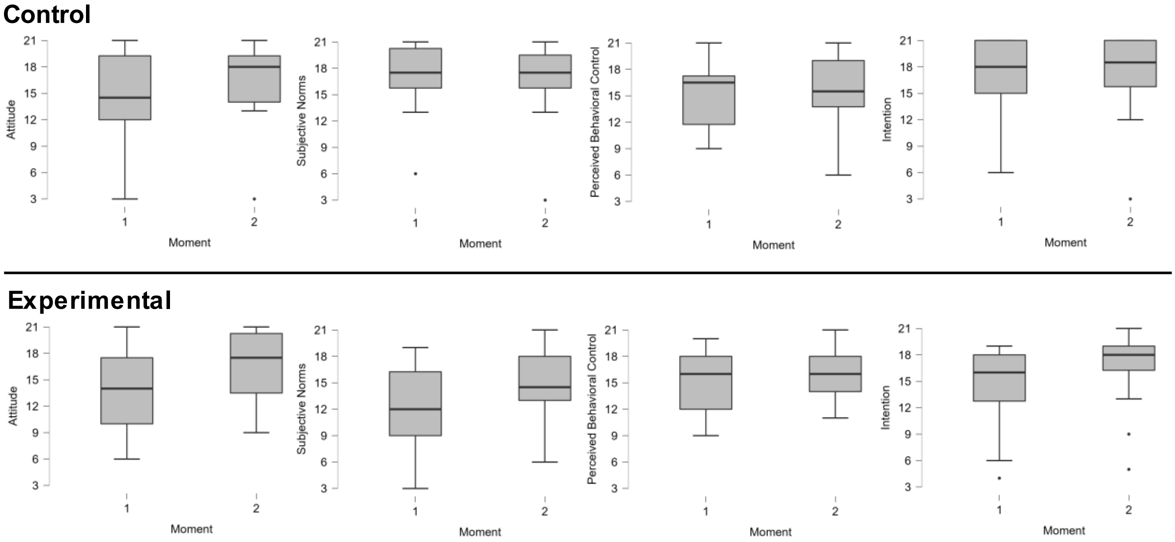


Figure 7.1 - Boxplots for Theory of Planned Behavior' variables. 1 = before the intervention; 2 = after.

Overall, the values for all variables improved, except for the control group's subjective norms and both groups' perceived behavioral control. Furthermore, regarding the choice of defense actions in pressure scenarios, the experimental group chose four out of five (median), and the control group chose three out of five. To verify the statistical significance of these results, we employed Null Hypothesis Significant Test (NHST), which represents the frequentist approach to data analysis. We tested normality for all variables using the Shapiro-Wilk test. In the within-group test, perceived behavioral control and intentions were not normal. In the between-groups test, attitudes, subjective norms (for the experimental group), intentions, and the number of chosen defense actions in scenarios were not normal. Therefore, we used the Wilcoxon Signed-Rank test for the paired sample (Hypotheses H1a-H1d) and the Mann-Whitney test for the independent samples (Hypotheses H2a-H2d and Hypothesis H5a). In Table Table 7.2, we present our results.

Table 7.2 - NHST Results.

Hypotheses	W	p-value	Rank-Biserial Correlation
H1a: attitudes in exp. group	90.0	0.047	0.500
H2a: subjective norms in exp. group	73.5	0.027	0.615
H3a: perceived behavioral control in exp. group	72.5	0.109	0.381
H4a: intentions in exp. group	91	0.008	0.733
H1b: attitudes between groups	124.5	0.560	-0.027

H2b: subjective norms between groups	82.5	0.959	-0.355
H3b: perceived behavioral control between groups	126.0	0.538	-0.016
H4b: intentions between groups	100.5	0.858	-0.215
H5b: scenarios between groups	168.5	0.052	0.326

As we explained in Section 7.2.3, we also carried out a Bayesian Hypothesis Testing (BHT) to compare the plausibility of the research and null hypotheses *relative to one another*. Figure 7.2 presents the results for the experimental group before and after they engaged with the digital simulation with the defense lenses. In the figure, BF_{+0} is the Bayes Factor for our research hypotheses (H1a-H1d), which predicted improvements in all variables. At the same time, BF_{0+} is the Bayes Factor for the null hypotheses representing no improvement. The Bayes Factor tells us the extent to which a hypothesis predicts the given data compared to others, providing a measure of the strength of evidence of one over the other [235]. The figure also presents a visual representation of the Bayes Factor through a pie chart—the larger the red area, the higher the support for the research hypotheses. The dotted density line represents the prior, which in our case was uninformative (a Cauchy distribution with a scale equal to 0.707), while the full density line presents the posterior. The gray dots represent the prior and posterior specific densities at the test value. When the gray dot of the posterior gets far below the one for the prior, we have higher support for the research hypothesis. The figure also shows the median of the effect size and its 95% confidence interval (CI). This is the effect size on a latent level (see more about this on van Doorn et al. [235]).

Figure 7.2 reveals that our research hypothesis regarding attitudes after exposure to our approach is a bit more probable than the null hypothesis (due to the BF_{+0} of 1.370). Therefore, although we have evidence in favor of H1a, it is weak due to a Bayes Factor between one and three—when considering the ranges of the strength of evidence of Erdogmus [235]. Figure 7.2 also shows that the intervention impacted subjective norms in the experimental group, with a BF_{+0} of 5.317, which is evidence of moderate strength in favor of H2a. We also have evidence of weak strength of an effect over perceived behavioral control in Figure 7.2, given the BF_{+0} of 1.394. Figure 7.2 indicates that the intervention also impacted intentions, with a BF_{+0} of 5.319, showing we have evidence of moderate strength in favor of H4a over the null hypothesis. Therefore, considering the Theory of Planned Behavior, these results indicate that within the experimental group, the proposed approach increased the intentions of defending software estimates when facing pressure, probably through improvements in all its antecedents, primarily through subjective norms.

Regarding the hypotheses comparing the control and experimental groups, we found weak to moderate evidence only for the null hypotheses for all the TPB variables, with BF_{0+} (which is the Bayes-Factor for the null hypotheses) varying from 3.01 for attitudes to 6.23 for subjective norms. Therefore, we found no support from evidence for H2a-H2d, as we found no evidence of higher scores for the experimental group. As for the response to pressure scenarios, we found a BF_{+0} of 1.219, which favors H5a but with evidence of weak strength. All the graphs for the testing regarding Hypotheses H2a-H2d and H5a are in Figure 7.3.

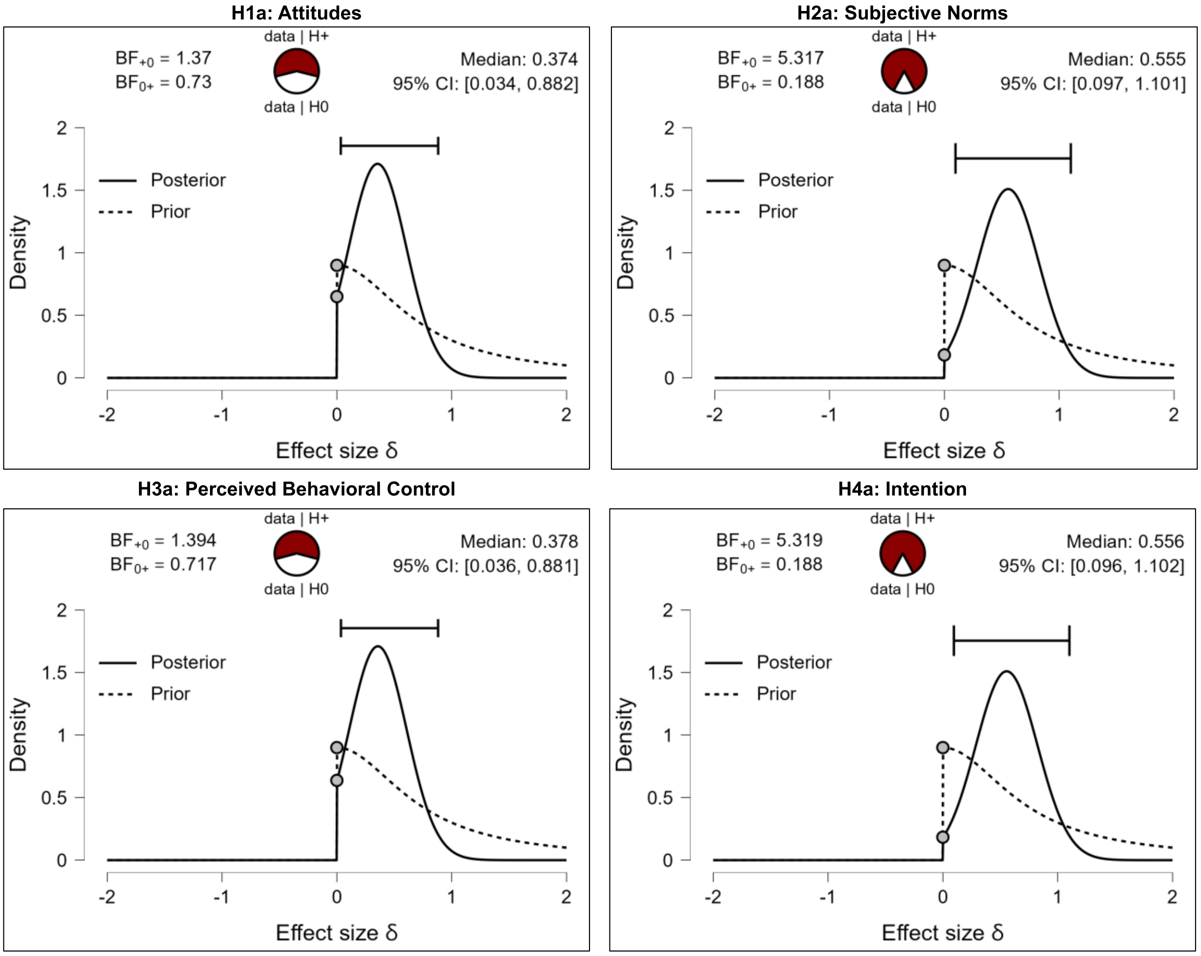


Figure 7.2 - Hypothesis Testing Results for the Experimental Group Before and After the Intervention.

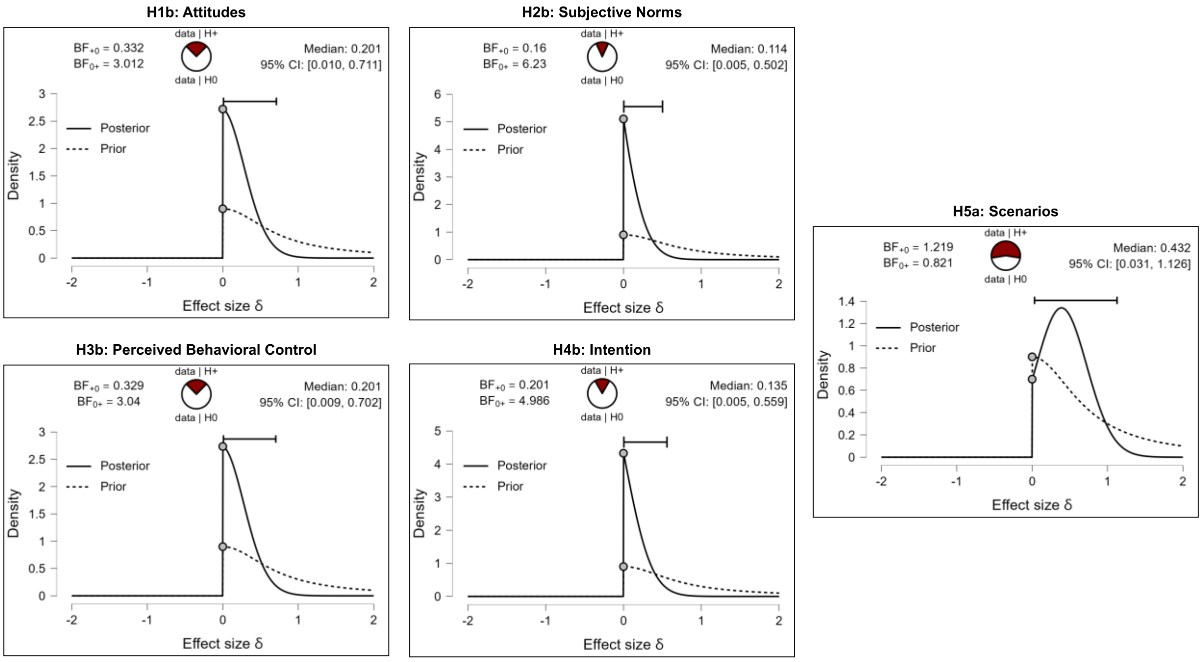


Figure 7.3 - Hypothesis Testing Results between the groups.

7.3.2. Perceived Usefulness

As part of the post-questionnaire, we asked participants of the experimental group about their perceived usefulness of learning negotiation principles encapsulated in the defense lenses as presented in the digital simulation. If they answered positively, we continued our investigation by asking them to explain which lenses or principles they believed to be the most useful for defending software estimates in their work and why. Otherwise, we asked them why the lenses would not be helpful.

All participants stated they found the principles can be useful in their work environments. Different participants mentioned different lenses/principles as the most useful ones. For instance, P5 chose the "Assert your Estimate" lens "*because it exercises the appreciation of the rationale that lies behind the estimate*". P18 picked the "Laddering Whys" lens as "*asking questions makes us reflect on motivations and difficulties of people involved in the projects, and I think this will be useful in my estimates.*" P29 chose the "Candidate Commitments" lens because "*in my experience, flexibility is really important in negotiations. Trying to find a commitment that brings benefits to all parties is hard, but I believe is the best alternative.*" All these lenses were from the Minimal Pack, but some participants also chose lens from the Extended Pack too. For example, P2 pointed the "Keep Strategy" lens as the most useful one "*because I always have reasons outside my control, so I can look for tools to defend* *my estimate.*” Another participant, P11 chose the “*Perspective Taking, because when we take a wider look, we can have arguments to base or to balance a point of view*.”

Collectively, participants mentioned all lenses, except one: “Choose your battles”. The most cited lens was the Candidate Commitments one. Moreover, some participants stated that all lenses are relevant because they complement each other, and each lens fits a different situation. In the words of P35: “*I think the principles were interesting. It is possible to balance time x quality. I believe the lenses complement each other, and knowing how to negotiate is part of the job.*” One participant also stated that both videos helped gain confidence to defend the software estimates. Next, we discuss these results in-depth.

7.4. Discussion

Are software engineers willing to move on from the software engineers’ gambit in the software development game? In the following sections we focus on this.

7.4.1. Comparing the Groups

We found weak to moderate evidence in favor of the null hypotheses of no increase in any of the TPB variables (intentions, attitudes, subjective norms and perceived behavioral control) for participants in the experimental group *compared with* participants in a control (Section 7.3.1). To understand more about this, we tested the difference in scores between these groups *before* the interventions. We found evidence of moderate strength for differences in subjective norms (BF_{10} = 5.010) and of weak strength for the intentions (BF_{10} = 1.390. The complete results of such tests are in APPENDIX G. This indicates that participants in the control group were possibly more inclined to defend their estimates in the first place. The evidence we have in favor of the proposed approach when comparing the two groups comes from the response to pressure scenarios: participants in the experimental group chose action defenses in one scenario more (median) than the control group. This is a positive answer to the question that opened this section, showing a higher willingness to adopt the strategy of defending software estimates when participants learned more about it.

Takeaway message: Exposing software practitioners to a digital simulation presenting the defense lenses with negotiation principles increases their chances of defending their estimates instead of yielding to unreasonable pressure.

7.4.2. Before and After the Intervention—Experimental Group

Another part of the answer to the question opening this section comes from the difference before and after the intervention, considering experimental group participants. We found weak evidence of higher scores for attitudes. Attitudes are a function of beliefs about a behavior’s likely consequences—their outcomes or experiences [41]. The evaluation of such outcomes—as desirable or not—also matters [56]. In our study, the digital simulation probably improved the participants' beliefs that performing the defense of estimates would lead to outcomes that participants regard favorably. This can include a better work experience, sufficient time to make higher-quality deliveries, and lower overtime work.

We also found evidence of higher scores for subjective norms, which refer to the person’s perceptions about (i) whether relevant referent individuals or groups approve (or disapprove) the behavior in question and (ii) whether such referents perform it [41]. The referent's importance to the person also plays a role [56]. So the digital simulation might have caused participants to think that colleagues, bosses, and clients would approve of defending estimates, given that it can protect product quality, the company image, and other of their interests.

It was surprising to find evidence of only weak strength for improvements in perceived behavioral control, as it is about beliefs regarding the presence of factors that can facilitate (or hinder) the behavior performance, including beliefs about skills [41]. This result was not a matter of participants misunderstanding the lenses: when discussing their perceived usefulness, participants demonstrated understanding correctly the principles the lenses embodied. One possible explanation relates to the measurement instrument: it might not have covered all relevant items for perceived behavioral control. We explore this issue in Section 7.5. Another explanation is that participants may need more time to exercise their newly acquired knowledge. Previous research on digital simulations suggested that they can make participants more aware of what they do not know, realizing their increasing skills as the training goes on for a more extended period [40]. Yet another possible explanation is that impeding control factors, such as lack of cooperation from other stakeholders, might play a large role over the control feelings regarding this specific behavior. Maybe the defense lenses learned through the digital simulation might look like a tool that does not directly affect such factors at first glance. However, the defense lenses are primarily about dealing with people unwilling to cooperate to define a realistic commitment based on estimates. Again, understanding this might require more experience using the lenses, possibly through longer learning periods.

Takeaway message: Exposing software practitioners to a digital simulation presenting the defense lenses with negotiation principles is very likely to improve their perceptions about subjective norms towards the behavior of defending their software estimates.

Regarding intentions, we found evidence of moderate strength of an increase in scores: a clear indication that people got more inclined to defend their estimates after the digital simulation, instead of to succumb to the software engineer's gambit. The stronger the intention, the more likely people will perform the behavior. However, it is no assurance as a varied set of factors can prevent people from acting on their intentions [41]. In any case, we have evidence that a short digital simulation—the two videos are no longer than 30 minutes together—is enough for people to grasp the ideas behind the set of lenses and positively change their intentions.

Qualitative analysis of the perceived usefulness of the lenses also provides evidence in favor of the applicability of the lenses in the field. Collectively, participants mentioned all the lenses as useful except for one: the "Choose your battles" lens. Interestingly, this lens aims to help estimators to act strategically, identifying the situations that keeping the estimate might not be in their best interest—or in other words, when yielding to pressure might be the best option.

Takeaway message: Exposing software practitioners to a digital simulation presenting the defense lenses with negotiation principles increases their intentions towards the behavior of defending their software estimates.

7.4.3. Before and After the Intervention—Control Group

Another side of our answer regards the difference before and after the reflection questions in the control group. We expected the reflection on the pressure scenarios and their impact on participants' lives, projects, and organizations could elicit the typical behavior people have when facing pressure over their estimates. Thus, we tested whether this simpler intervention of reflection, which takes much less to participate in than the digital simulation, would also increase scores for intentions. We executed the same analysis procedures we did for the experimental group. We found weak evidence in favor of improvements only for attitudes (BF_{+0} of 2.403). This shows that the reflection also makes evident how good and useful the defense of estimates is, possibly through remembering how bad pressure outcomes are. However, the evidence favored the null hypotheses for all other variables, as Section 7.3.1 fully reports. Furthermore, descriptive statistics suggest a drop in perceived behavioral control. Therefore, while attitudes increased through reflection, the feeling of the capacity to perform the behavior decreased.

Takeaway message: Exposing software practitioners to a reflection on pressure scenarios and their outcomes is likely to improve attitudes towards the defense of software estimates and to worsen perceived behavioral control, leading to no increase in intentions of performing such behavior.

7.4.4. Cost x Benefit Analysis of the Proposed Approach

The final issue to discuss about moving on from the software engineers' gambit is the practical significance of our results [236]: the core concern in supporting the choice of concrete actions by practitioners in the software industry. After all, is it worth engaging with the digital simulation and the study of the defense lenses? To answer that, we need to analyze the costs and benefits of doing so. Starting with the **cost**, it takes less than 30 minutes to participate in the digital simulation. For the ones interested in reading the booklet with the defense lenses, it can take up to 30 minutes more: approximately 13 minutes for the main text and one minute and a half for each lens. We made this estimation using https://thereadtime.com/ for silent reading (around 238 words per minute).

Assessing the **benefits** is not so straightforward. First, what are the expected outcomes of an increased intention to defend realistic software estimates, considering this is the main result of participating in the digital simulation? We expect it to increase defense behavior, but it remains unclear how much an increase in intention is necessary to secure that. In any case, when asked whether they think learning negotiation principles is useful for defending software estimates in their current work environment, all participants of the experimental group answered positively. This can be a rough indication that such people will follow on their intentions to defend their estimates. As revealed in the words of one of the participants (P22), sent to us through an e-mail after participation in the experimental group: "*The lenses are broad concepts and have good application in the real world. And I think they are applicable in any relation of estimates x delivery. (...) I am interested in going deeper on this and delivering this content to my manager. He is the one who gets our estimates and presents them to the client. (...) It would also be nice to apply the lenses when developers are estimating.*"

Second, if an increase in defense behavior does materialize, what are the consequences for individuals and companies? The qualitative analysis of answers to the reflection questions from the control group can give us some hints on this. Participants mentioned various outcomes from pressure. Some were related to the product or the process, such as an increase in product failure/bugs leading to lack of trust in the product, product instability, neglect of long-run maintenance and testing activities, neglect of good practices, and overall lower quality. Some other outcomes were related to the client, such as unmet expectations and needs. Other outcomes were related to the team and estimators: overtime work, emotional distress, resignation, and solution block. We present quotations supporting each of these outcomes in APPENDIX G. Improvements in any of these outcomes can benefit individual practitioners and their companies.

Takeaway message: The digital simulation and the study of the defense lenses is a low-time intervention with the potential to impact varied outcomes from pressure related to product/process quality, the client's needs, and the software practitioners' quality of life.

7.5. Threats to Validity

A threat to conclusion validity in our study regards the reliability of measures. We performed a reliability analysis for TPB items after the first moment of data collection (pre-questionnaire, 45 data points). We got acceptable reliability scores (Cronbach's α higher than 0.7) for subjective norms and intentions. We got lower values for attitudes (Cronbach's α of 0.66) and perceived behavioral control (Cronbach's α of 0.33). For attitudes, dropping one item resolves the issue. Dropping one item also improves subjective norms' and perceived behavioral control' Cronbach's α. But to improve it further, we would need to add more items to the questionnaire and another step of data collection with the new questionnaire to compare answers before and after the interventions. This would require more time from participants in our study, potentially increasing mortality because software practitioners have a short time to devote to participation in research. As we valued keeping a sample of participants who were active software practitioners, increasing the relevance of our results, we decided to keep the questionnaire untouched during data collection. We dropped one item for each intention's antecedents during the final analysis, increasing their Cronbach's α.

A threat to internal validity in our study regards mortality, as we had participants dropping out after the first moment of data collection: six in the experimental group and seven in the control group. Therefore, we compared dropouts to participants regarding demographic and TPB variables. We found significant differences in years of experience in software development and maintenance: dropouts were more experienced on average. We do not consider this an issue, as our proposal will likely benefit more inexperienced people.

Regarding construct validity, participants might not be familiarized with what is a defensive behavior toward software estimates. Moreover, we needed to design valid questions for assessing the TPB variables. Therefore, we constructed the questions about TPB with the guidance of a manual [61], which also required us to define to participants what is a defensive behavior of software estimates. We also piloted the questionnaires with four active software practitioners and improved the wording of questions to increase validity.

Controlled experiments, like ours, are generally limited in the number of subjects [207] and do not necessarily support statistical generalization [237]. Also, as typical in experiments, we prioritized internal over external validity. We valued assessing whether negotiation principles present in the defense lenses and the digital simulation could cause software practitioners to raise intentions in defending their software estimates over generalizing our results to a larger population. The positive qualitative feedback we received from participants in the digital simulation can be seen as a preliminary sign of its external applicability in the software industry.

7.6. Summary

In this study, we evaluated a new strategy for practitioners to move on from the software engineers' gambit, in which they sacrifice quality—of products and of life—to gain time due to pressure over their software estimates. The proposed strategy is comprised of a set of lenses to support software estimators in defending software estimates during their communication to other relevant stakeholders. The lenses embed the principles of consolidated negotiation methods, thus also contributing to a concrete approach toward increasing negotiation skills among software practitioners. Moreover, we presented the defense lenses in a lightweight digital simulation format, making the acquisition of their knowledge more dynamic, low-cost, and requiring low time—important features for dissemination among already too-pressed software practitioners, who are used to play the software engineers' gambit.

In addition, we provide supporting evidence that engaging with the digital simulation and learning the defense lenses increases participants' intentions in defending software estimates when facing pressure and their attitudes and perceptions of how good such behavior is from the perspective of other relevant people in their context. It also increases their choices of defense actions in comparison with a control group, revealing they are more likely to stop yielding to pressure. Furthermore, qualitative evidence shows that participants exposed to our approach found the principles can be useful in their daily industrial practice, further revealing the relevance of this work. Currently, we provided the booklet and the digital simulation for all participants in the study, including the ones in the control group. As part of our future work, we plan to follow up with participants to understand whether they applied the lenses in their work environments and their perception of their usefulness in the wild.

CHAPTER 8 – CONCLUSIONS

This chapter presents our final considerations. We overview the steps we performed in the path to designing our proposed artifact. Finally, we provide an overview of the contribution of this work.

8.1. Final Considerations

In this dissertation, we discuss how to help estimators to move on from the software engineers' gambit to a defense of their software estimates, thus deflecting from imposed and unfeasible commitments whenever possible. A game theoretical perspective of software development, acknowledging it involves different players looking for the best strategies to increase their payoffs, led us to the study of an underexplored strategy to deal with pressure from stakeholders over software estimates: the use of negotiation principles adapted to the estimation context. The effective use of these principles and the adoption of a defense strategy requires learning and behavior change, which we addressed with the design of an artifact focused on supporting software practitioners dealing with pressure. Our approach focuses on the behavioral side of software effort estimation, without contributions on how to make a technically sound estimate. It is applicable in cases where the estimator is confident that they have a good estimate for the task/project and are ready to communicate it to other stakeholders to make a commitment.

By using the DSR methodology, we conducted a series of empirical studies to create the proposed artifact rigorously. Reflecting on the closure of this DSR project, we can say we had two major phases of knowledge acquisition. The first phase was before devising the first version of our artifact. At this phase, knowledge acquisition focused mostly on problem investigation. The first empirical studies were a Systematic Literature Mapping (SLM) and a qualitative study providing perspectives from academy and practice, respectively. These studies helped us to grasp the need for improvement and informed us the problematic scenarios in which our artifact could be useful. This phase also focused on the gain of knowledge to deal with the identified problem. The literature review on negotiation provided this.

The second phase of knowledge acquisition focused on understanding the artifact' perceived usefulness and improvement opportunities, enabling its refinement. Two empirical studies supported this phase: a focus group and a controlled experiment. Both studies involved participants who are active software professionals.

The evaluation results provide evidence that practitioners can learn the lenses through the digital simulation and perceive them as useful for application in their current work environments. Moreover, different participants mentioned different lenses as useful, indicating that the set of existing pressure scenarios is varied throughout software organizations. Therefore, we recommend practitioners to use the digital simulation to learn on the whole set of lenses, further studying them by reading the booklet. Next, practitioners can revisit the booklet whenever they are preparing for an estimation session or are about to communicate their estimates to a stakeholder. In doing so, they can focus on a few cards that they think are more useful for their specific context, thus strengthening their skills on the negotiation principles supporting such lenses—and thus boosting the abilities to resist the kinds of pressure that are typical in their working environments.

Next, we discuss the publications resulting from the studies we conducted and their individual contributions. We also discuss the overall contributions from this work as a whole. Moreover, we discuss the future work we plan to carry out to strengthen the proposed approach and disseminate it, as well as ramifications of this work related to Behavioral Software Engineering.

8.2. Contributions and Publications

Throughout this work, we made several contributions and published papers to disseminate our results through the Software Engineering community. Table 8.1 presents the publications that resulted from the steps that we executed as part of this research project. We also discuss how the papers we wrote and published connect to the chapters of this text and when we executed each study, to provide an idea of the timeline for the production of this dissertation.

Table 8.1 - Publications and contributions

Written and published papers	
Publication	**Contribution**
Research proposal presented in IDoESE 2019, detailing the research objectives and methodology, published at Software Engineering Notes [22].	Description of a research methodology to investigate the use of negotiation theories and techniques to improve software project estimation.
SLM about factors affecting expert judgment estimates, published at the Journal of Systems and Software [5].	• A map organizing the factors affecting expert judgment estimates by project phase, stakeholder, and type of effect.

Written and published papers	
Publication	**Contribution**
	• Identification of the research and measurement strategies that researchers employed the most.
SLM about factors affecting overconfidence and uncertainty assessments, published at the Brazilian Simposium of Software Engineering (SBES) 2021 – Research Track [238].	• Identification of factors affecting overconfidence and uncertainty assessment in software estimation. • Identification of the research and measurement strategies that researchers employed the most.
Exploration of the SLM factors, depicting software effort estimation as more than a prediction task: as a behavioral act. Submitted to the Empirical Software Engineering Journal.	• Description of two latent themes, representing perspectives on the factors affecting software estimates: as a technical prediction task and as a behavioral act. • Presentation and evaluation of strength of evidence of review findings regarding the factors associated with the behavioral act theme.
Qualitative study about how software practitioners use estimates to establish commitments, published at the International Conference on Cooperative and Human Aspects of Software Engineering (CHASE) 2021 [8].	Evidence for: • Changes in software estimates to make them defensible. • Three different reasons for padding in the software industry: (i) contingency buffer, (ii) completing other tasks, (iii) improving overall quality. • Padding as a tool for balancing short and long-term needs in software development and maintenance.
Description of the research problem and our proposed solution, with the results from the focus group, published at the International Conference on Software Engineering – New Ideas and Emerging Results Track (ICSE NIER) 2022 [21].	• Description of the research problem proposition, the negotiation theories, and one of the defense lenses. • Empirical evidence on the lenses' perceived usefulness and improvement opportunities.
Complete description of the artifact (defense lenses and digital simulation) and of the controlled experiment. Accepted for publication at the International Conference on Software Engineering – Technical Track 2023.	• Description of the digital simulation and the defense lenses, their theoretical background, and their rationale. • Empirical evidence about the intentions of software practitioners to adopt the defense lenses. • Empirical evidence on the digital simulation's and lenses' perceived usefulness.

First, we presented our research proposal at IDoESE (International Doctoral Symposium on Empirical Software Engineering) 2019 [22]. We described our research problem and the approach that we envisioned, together with our research methodology—which corresponds to a lot of what we discussed in CHAPTER 1.

Next, we described the studies that we carried out to understand deeper our research problem, namely an SLM and a qualitative study in the software industry. We executed the SLM from the end of the year 2018 to the mid of the year 2020. The results from the SLM showed how pressure over the estimates is among the many relevant factors affecting expert judgment estimates and how removing padding from the estimates can harm their accuracy. We explored these results in CHAPTER 2 and wrote a paper about it, accepted for publication at the Journal of Systems and Software in the end of 2021.

Moreover, our SLM inspired us to execute a secondary SLM about the factors affecting overconfidence and the uncertainty assessment of estimates, specific aspects of estimates' accuracy. We found a total of eight factors. We published it at the SBES (Simpósio Brasileiro de Engenharia de Software - Brazilian Symposium on Software Engineering) 2021. We did not include the paper results in this dissertation because it is not entirely aligned with our proposed research question.

We also submitted an additional paper to the Empirical Software Engineering Journal, with an exploration of the SLM factors. This paper depicts software effort estimation as more than a prediction task: as a behavioral act. We also did not include the paper results in this dissertation because it is not entirely aligned with our proposed research question.

We executed the qualitative study from the mid of 2019 to the mid of 2020. Its results reinforced the evidence on how pressure affects estimates, leading estimators to change their estimates to make them defensible and to use padding as a tool to balance short and long-term needs in software development and maintenance [8]. Therefore, padding is a strategy for dealing with pressure in the software industry. We explored these results in CHAPTER 4 and published them at CHASE (Cooperative and Human Aspects of Software Engineering) 2021 [8].

This dissertation also described the solution that we propose together with its theoretical foundations. We resorted to negotiation methods introduced in CHAPTER 2, looking for an alternative that could allow estimators to defend their estimates and negotiate effectively for realistic commitments instead of yielding to pressure, changing their estimates, or pad other tasks to compensate. Therefore, we devised the defense lenses as CHAPTER 5 explains. We worked on the first version of our artifact from the mid of the year 2019 to the beginning of the year 2021.

Next, CHAPTER 6 presents the details of a focus group to evaluate our proposed approach. Our evaluation strategy relied on experts' opinions on the usefulness and usability of the defense lenses. We worked on it for about three months starting at the mid of the year 2021. We described the defense lenses and the results from the focus group in a paper published at the International Conference on Software Engineering – New Ideas and Emerging Results Track (ICSE NIER) 2022.

Right after the focus group, we worked on refining the artifact, by improving the defense lenses and crafting the digital simulation in line with the improvement opportunities revealed in the focus group. We evaluated the artifact with a controlled experiment. We started planning the controlled experiment at the mid of the year 2021. We finished its data collection and analysis in the mid of the year 2022. CHAPTER 7 presents the results of this study, accepted for publication at the International Conference on Software Engineering – Technical Track 2023.

In addition to the individual contributions of each chapter and publication abovementioned, we also provide additional contributions to the Software Engineering community. First, the map of factors affecting expert-judgment software estimates resulting from our SLM is comprehensive enough to benefit other researchers investigating topics related to software effort estimation. Second, our approach incorporates principles from negotiation, thus holding the potential to promote the learning of this soft skill among software practitioners. Third, we propose an approach to deal with the behavior of stakeholders, including estimators—a step towards Behavioral Software Engineering, i.e., to the "study of cognitive, behavioral and social aspects of software engineering performed by individuals, groups or organizations" [38]. Fourth, we incorporated to our analysis the Theory of Planned Behavior, a consolidated social science theory, showing how it can be useful to understand behavior change in our field.

8.3. Future Work

One issue that arises from our evaluation studies (the focus group and the controlled experiment) is that they provide evidence on the perceived usefulness and intentions of use of our proposed approach from the practitioners' perspective. However, we still need evidence on the use of the lenses in real projects in the software industry. When using the lenses in such projects, practitioners might face additional difficulties and identify other improvement opportunities for the lenses or the digital simulation. Therefore, as part of our future work, we plan to **follow up with participants from the controlled experiment** to understand whether they applied the lenses in their work environments and their perception of their usefulness in the wild. Currently, we provided the booklet and the digital simulation for all participants in the study, including the ones in the control group.

With the knowledge gained from following up participants in the industry, we also plan to collect more data on the lenses usage scenarios and **automate a flow of lenses' recommendations to practitioners**. With this, we expect to make it easier for them to understand which lenses are applicable to which scenarios.

As the software industry gets more in need of software engineers with enhanced software skills, another future endeavor is to **adapt and evaluate the approach for teaching software engineering students**. We can start with a study based on the current approach and work towards improving it for this new context. We can also **explore whether the negotiation knowledge gained with our proposed approach can spill over to other Software Engineering knowledge areas where they can be useful**, such as Requirements Engineering.

Finally, we intend to **explore further the behavioral side of software estimation**. Our current approach represents a boost intervention [39] but there is a whole class of other types of behavioral interventions to explore such as choice architecture techniques, also known as nudges [239], and noise reduction strategies [66]. Some of these techniques are explored in the software effort estimation literature. For instance, relative estimation employs the overall idea of anchored rating scales, which involve creating a better rating format by devising a scale that establishes a common frame of reference [239]. After all, relative estimation is about choosing a reference case (a story or a task), estimating it by giving a certain amount of story points, and then estimating the remaining cases by comparing with it [209], [240]. However, some of these choice architecture and noise reduction techniques remain unexplored. For instance, we could benefit from integrating frame of reference training, which is about training raters to recognize different dimensions, to anchored rating scales. Perhaps, companies and teams can use a few anchor stories, tasks, or requirements to support the training of their estimators. It can be complex and time-consuming, requiring customization to the organization/unit and constant update of cases [66]. However, noisy and biased estimation is also costly.

REFERENCES

[1] "Gambit - Chess Terms," *Chess.com*. https://www.chess.com/terms/chess-gambit (accessed Jul. 06, 2022).

[2] M. Vidoni, Z. Codabux, and F. H. Fard, "Infinite technical debt," *Journal of Systems and Software*, vol. 190, p. 111336, Aug. 2022, doi: 10.1016/j.jss.2022.111336.

[3] C. Gavidia-Calderon, F. Sarro, M. Harman, and E. T. Barr, "Game-theoretic analysis of development practices: Challenges and opportunities," *Journal of Systems and Software*, vol. 159, p. 110424, Jan. 2020, doi: 10.1016/j.jss.2019.110424.

[4] M. Grechanik and D. Perry, "Analyzing software development as a noncooperative game," in *26th International Conference on Software Engineering - Sixth International Workshop on Economics-Driven Software Engineering Research (EDSER-6)*, Edinburgh, UK, Jan. 2004, pp. 29–34. doi: 10.1049/ic:20040282.

[5] P. Matsubara, B. Gadelha, I. Steinmacher, and T. Conte, "SEXTAMT: A systematic map to navigate the wide seas of factors affecting expert judgment software estimates," *The Journal of Systems and Software*, vol. 185, p. 111148, 2022, doi: 10.1016/j.jss.2021.111148.

[6] C. Jones, "Social and Technical Reasons for Software Project Failures," *Crosstalk*, vol. 19, no. 6, 2006.

[7] A. Magazinius, S. Börjesson, and R. Feldt, "Investigating intentional distortions in software cost estimation – An exploratory study," *Journal of Systems and Software*, vol. 85, no. 8, pp. 1770–1781, Aug. 2012, doi: 10.1016/j.jss.2012.03.026.

[8] P. Matsubara, I. Steinmacher, B. Gadelha, and T. Conte, "Buying time in software development: how estimates become commitments?," in *Proceedings of the 14th International Conference on Cooperative and Human Aspects of Software Engineering*, Madrid, Spain, 2021, pp. 61–70.

[9] M. Lavallée and P. N. Robillard, "Why Good Developers Write Bad Code: An Observational Case Study of the Impacts of Organizational Factors on Software Quality," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, May 2015, vol. 1, pp. 677–687. doi: 10.1109/ICSE.2015.83.

[10] R. E. Fairley and M. J. Willshire, "Why the vasa sank: 10 problems and some antidotes for software projects," *IEEE Softw.*, vol. 20, no. 2, pp. 18–25, Mar. 2003, doi: 10.1109/MS.2003.1184161.

[11] M. Kuutila, M. Mäntylä, U. Farooq, and M. Claes, "Time pressure in software engineering: A systematic review," *Information and Software Technology*, vol. 121, p. 106257, May 2020, doi: 10.1016/j.infsof.2020.106257.

[12] M. Kuutila, M. V. Mantyla, U. Farooq, and M. Claes, "What Do We Know About Time Pressure in Software Development?," *IEEE Software*, vol. 38, no. 5, pp. 32–38, Sep. 2021, doi: 10.1109/MS.2020.3020784.

[13] D. Girardi, F. Lanubile, N. Novielli, and A. Serebrenik, "Emotions and Perceived Productivity of Software Developers at the Workplace," *IEEE Transactions on Software Engineering*, pp. 1–1, 2021, doi: 10.1109/TSE.2021.3087906.

[14] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "What happens when software developers are (un)happy," *Journal of Systems and Software*, vol. 140, pp. 32–47, Jun. 2018, doi: 10.1016/j.jss.2018.02.041.

[15] D. Hooper and K. Whyld, *The Oxford companion to chess*, 2nd ed. Oxford University Press, 1992.

[16] F. Brooks, *Mythical Man-Month, The: Essays on Software Engineering, Anniversary Edition*, Anniversary edition. Reading, Mass: Addison-Wesley Professional, 1995.

[17] S. McConnell, "Politics, Negotiation, and Problem Solving," in *Software Estimation: Demystifying the Black Art*, Redmond: Microsoft Press, 2006, pp. 259–270.

[18] R. Fisher, W. Ury, and B. Patton, *Getting to Yes: Negotiating Agreement Without Giving in*, 3rd Revised. New York, NY, USA: Penguin Books, 2011.

[19] W. Ury, *Getting Past No: Negotiating in Difficult Situations*, Revised. New York, NY, USA: Bantam, 2007.

[20] W. Ury, *The Power of A Positive No*. London: Hodder & Stoughton, 2012.

[21] P. Matsubara, I. Steinmacher, B. Gadelha, and T. Conte, "The best defense is a good defense: adapting negotiation methods for tackling pressure over software project estimates," in *Proceedings of the 44th International Conference in Software Engineering*, Pittsburgh, Pennsylvania, 2022. doi: 10.1145/3510455.3512775.

[22] P. G. Fernandes Matsubara, "Dealing with software estimates distortions from the perspective of negotiation theories," *SIGSOFT Softw. Eng. Notes*, vol. 44, no. 3, p. 22, Nov. 2019, doi: 10.1145/3356773.3356794.

[23] R. J. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*. Berlin Heidelberg: Springer-Verlag, 2014. doi: 10.1007/978-3-662-43839-8.

[24] R. Wieringa, "Design Science As Nested Problem Solving," in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, New York, NY, USA, 2009, p. 8:1-8:12. doi: 10.1145/1555619.1555630.

[25] A. R. Hevner, "A Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems*, vol. 19, no. 2, p. 7, 2007.

[26] T. Halkjelsvik and M. Jørgensen, "Overoptimistic Predictions," in *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*, T. Halkjelsvik and M. Jørgensen, Eds. Cham: Springer International Publishing, 2018, pp. 35–54. doi: 10.1007/978-3-319-74953-2_4.

[27] T. Halkjelsvik and M. Jørgensen, "Time Prediction Biases," in *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*, T. Halkjelsvik and M. Jørgensen, Eds. Cham: Springer International Publishing, 2018, pp. 55–70. doi: 10.1007/978-3-319-74953-2_5.

[28] J. Lande, "Taming the Jungle of Negotiation Theories," Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 3089855, Dec. 2017. Accessed: Apr. 23, 2019. [Online]. Available: https://papers.ssrn.com/abstract=3089855

[29] M. Roloff, L. Putnam, and L. Anastasiou, "Negotiation Skills," in *Handbook of Communication and Social Interaction Skills*, Lawrence Erlbaum Associates, 2003, pp. 801–833.

[30] J. Brett and L. Thompson, "Negotiation," *Organizational Behavior and Human Decision Processes*, vol. 136, pp. 68–79, Sep. 2016, doi: 10.1016/j.obhdp.2016.06.003.

[31] K. Molokken and M. Jorgensen, "A review of software surveys on software effort estimation," in *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, Sep. 2003, pp. 223–230. doi: 10.1109/ISESE.2003.1237981.

[32] A. Trendowicz, J. Münch, and R. Jeffery, "State of the Practice in Software Effort Estimation: A Survey and Literature Review," in *Software Engineering Techniques*, Berlin, Heidelberg, 2011, pp. 232–245.

[33] M. Usman, E. Mendes, and J. Börstler, "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*, Nanjing, China, 2015, pp. 1–10. doi: 10.1145/2745802.2745813.

[34] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra, "Research patterns and trends in software effort estimation," *Information and Software Technology*, vol. 91, pp. 1–21, Nov. 2017, doi: 10.1016/j.infsof.2017.06.002.

[35] S. Deterding, "The lens of intrinsic skill atoms: A method for gameful design," *Human-Computer Interaction*, vol. 30, no. 3–4, pp. 294–335, 2015, doi: 10.1080/07370024.2014.993471.

[36] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A Model for Technology Transfer in Practice," *IEEE Softw.*, vol. 23, no. 6, pp. 88–95, Nov. 2006, doi: 10.1109/MS.2006.147.

[37] J. Kontio, J. Bragge, and L. Lehtola, "The Focus Group Method as an Empirical Tool in Software Engineering," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. London: Springer, 2008, pp. 93–116. doi: 10.1007/978-1-84800-044-5_4.

[38] P. Lenberg, R. Feldt, and L. G. Wallgren, "Behavioral software engineering: A definition and systematic literature review," *Journal of Systems and Software*, vol. 107, pp. 15–37, Sep. 2015, doi: 10.1016/j.jss.2015.04.084.

[39] R. Hertwig and T. Grüne-Yanoff, "Nudging and Boosting: Steering or Empowering Good Decisions," *Perspect Psychol Sci*, vol. 12, no. 6, pp. 973–986, Nov. 2017, doi: 10.1177/1745691617702496.

[40] A. Gegenfurtner, C. Quesada-Pallarès, and M. Knogler, "Digital simulation-based training: A meta-analysis," *British Journal of Educational Technology*, vol. 45, no. 6, pp. 1097–1114, 2014, doi: 10.1111/bjet.12188.

[41] I. Ajzen, "The theory of planned behavior: Frequently asked questions," *Human Behavior and Emerging Technologies*, vol. 2, no. 4, pp. 314–324, 2020, doi: 10.1002/hbe2.195.

[42] Working Group for Life Cycle Processes, "IEEE 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes." ISO/IEC/IEEE, 2017. Accessed: May 27, 2019. [Online]. Available: https://standards.ieee.org/standard/12207-2017.html

[43] S. McConnell, "What is an 'Estimate'?," in *Software Estimation: Desmystifying the Black Art*, Redmond: Microsoft Press, 2006, pp. 3–14.

[44] S. McConnell, "Flow of Software Estimates on a Well-Estimated Project," in *Software Estimation: Desmystifying the Black Art*, Redmond: Microsoft Press, 2006, pp. 171–180.

[45] D. Fairley, "Making accurate estimates," *IEEE Software*, vol. 19, no. 6, pp. 61–63, Nov. 2002, doi: 10.1109/MS.2002.1049392.

[46] A. Dagnino, "Estimating software-intensive projects in the absence of historical data," in *2013 35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, USA, May 2013, pp. 941–950. doi: 10.1109/ICSE.2013.6606643.

[47] A. K. Schneider, N. Ebner, D. Matz, and J. Lande, "The Definition of Negotiation: A Play in Three Acts," *Missouri Journal of Dispute Resolution*, vol. 1, p. 4, 2017, doi: 10.2139/ssrn.2902743.

[48] W. Steinel and F. Harinck, "Negotiation and Bargaining," *Oxford Research Encyclopedia of Psychology*. Sep. 28, 2020. doi: 10.1093/acrefore/9780190236557.013.253.

[49] F. R. Hak and K. Sanders, "Principled negotiation: an evidence-based perspective," *Evidence-based HRM: a Global Forum for Empirical Scholarship*, vol. 6, no. 1, pp. 66–76, Jan. 2018, doi: 10.1108/EBHRM-03-2017-0014.

[50] C. Menkel-Meadow, "Why Hasn't the World Gotten to Yes? An Appreciation and Some Reflections," *Negotiation Journal*, vol. 22, no. 4, pp. 485–503, 2006, doi: 10.1111/j.1571-9979.2006.00119.x.

[51] C.-J. Tsay and M. H. Bazerman, "A Decision-Making Perspective to Negotiation: A Review of the Past and a Look to the Future," *Negotiation Journal*, vol. 25, no. 4, pp. 467–480, 2009, doi: 10.1111/j.1571-9979.2009.00239.x.

[52] J. L. Sinskey, J. M. Chang, G. S. Shibata, A. J. Infosino, and K. Rouine-Rapp, "Applying Conflict Management Strategies to the Pediatric Operating Room," *Anesth Analg*, vol. 129, no. 4, pp. 1109–1117, Oct. 2019, doi: 10.1213/ANE.0000000000003991.

[53] D. Ames, A. Lee, and A. Wazlawek, "Interpersonal assertiveness: Inside the balancing act," *Social and Personality Psychology Compass*, vol. 11, no. 6, p. e12317, 2017, doi: 10.1111/spc3.12317.

[54] S. Eisen Jr, "Practical Guide to Negotiating in the Military," Air University Press, Technical Report AD1117925, Dec. 2019. Accessed: Oct. 14, 2021. [Online]. Available: https://apps.dtic.mil/sti/citations/AD1117925

[55] T. Lorey, P. Ralph, and M. Felderer, "Social Science Theories in Software Engineering Research," in *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)*, May 2022, pp. 1994–2005. doi: 10.1145/3510003.3510076.

[56] M. P. H. Kan and L. R. Fabrigar, "Theory of Planned Behavior," in *Encyclopedia of Personality and Individual Differences*, V. Zeigler-Hill and T. K. Shackelford, Eds. Cham: Springer International Publishing, 2017, pp. 1–8.

[57] I. Ajzen, "The Theory of Planned Behavior," in *Handbook of Theories of Social Psychology: Volume 1*, 1 vols., London: SAGE Publications Ltd, 2012, pp. 438–459. doi: 10.4135/9781446249215.

[58] M. B. Aulbach *et al.*, "A dual process model to predict adolescents' screen time and physical activity," *Psychology & Health*, vol. 0, no. 0, pp. 1–20, Oct. 2021, doi: 10.1080/08870446.2021.1988598.

[59] B. Gardner and C. Abraham, "Going Green? Modeling the Impact of Environmental Concerns and Perceptions of Transportation Alternatives on Decisions to Drive," *Journal of Applied Social Psychology*, vol. 40, no. 4, pp. 831–849, 2010, doi: 10.1111/j.1559-1816.2010.00600.x.

[60] R. Cooke, M. Dahdah, P. Norman, and D. P. French, "How well does the theory of planned behaviour predict alcohol consumption? A systematic review and meta-analysis," *Health Psychol Rev*, vol. 10, no. 2, pp. 148–167, Apr. 2016, doi: 10.1080/17437199.2014.947547.

[61] J. Francis *et al.*, "Constructing questionnaires based on the theory of planned behaviour: A manual for health services researchers," 2004. https://openaccess.city.ac.uk/id/eprint/1735/ (accessed Apr. 07, 2022).

[62] A. Magazinius and R. Feldt, "Confirming Distortional Behaviors in Software Cost Estimation Practice," in *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, Aug. 2011, pp. 411–418. doi: 10.1109/SEAA.2011.61.

[63] S. F. Ochoa, J. A. Pino, and F. Poblete, "Estimating Software Projects Based On Negotiation," *Journal of Universal Computer Science*, vol. 15, pp. 1812–1832, 2009.

[64] E. Løhre and M. Jørgensen, "Numerical anchors and their strong effects on software development effort estimates," *Journal of Systems and Software*, vol. 116, pp. 49–56, Jun. 2016, doi: 10.1016/j.jss.2015.03.015.

[65] T. Halkjelsvik and M. Jørgensen, "From origami to software development: A review of studies on judgment-based predictions of performance time.," *Psychological Bulletin*, vol. 138, no. 2, pp. 238–271, 2012, doi: 10.1037/a0025996.

[66] D. Kahneman, O. Sibony, and C. R. Sunstein, *Noise: A Flaw in Human Judgment*, 1st ed., vol. 1. New York: Little, Brown Spark, 2021.

[67] T. Halkjelsvik and M. Jørgensen, "How We Predict Time Usage," in *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*, T. Halkjelsvik and M. Jørgensen, Eds. Cham: Springer International Publishing, 2018, pp. 5–11. doi: 10.1007/978-3-319-74953-2_2.

[68] D. Kahneman, O. Sibony, and C. R. Sunstein, "Crime and noisy punishment," in *Noise: A Flaw in Human Judgment*, New York: Little, Brown Spark, 2021, pp. 18–25.

[69] D. Kahneman, O. Sibony, and C. R. Sunstein, "Sequencing information in foresinc science," in *Noise: A Flaw in Human Judgment*, New York: Little, Brown Spark, 2021, pp. 210–220.

[70] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, 1st ed. CRC Press, 2015.

[71] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, Aug. 2015.

[72] P. Matsubara, B. Gadelha, I. Steinmacher, and T. Conte, "Supplementary material for the SEXTAMT," *figshare*, Apr. 29, 2021. https://doi.org/10.6084/m9.figshare.14502405.v2 (accessed Apr. 29, 2021).

[73] C. Hugh, *The Encyclopaedia Britannica - Dictionary of arts, sciences, literature and general information*, 11th ed., vol. 24, 29 vols. 1911.

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

[75] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Information and Software Technology*, vol. 53, no. 6, pp. 625–637, Jun. 2011, doi: 10.1016/j.infsof.2010.12.010.

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

[77] C. Wohlin, "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*, New York, NY, USA, 2014, p. 38:1-38:10. doi: 10.1145/2601248.2601268.

[78] M.-A. Storey, N. A. Ernst, C. Williams, and E. Kalliamvakou, "The who, what, how of software engineering research: a socio-technical framework," *Empir Software Eng*, vol. 25, no. 5, pp. 4097–4129, Sep. 2020, doi: 10.1007/s10664-020-09858-z.

[79] M. Shepperd, C. Mair, and M. Jørgensen, "An Experimental Evaluation of a De-biasing Intervention for Professional Software Developers," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2018, pp. 1510–1517. doi: 10.1145/3167132.3167293.

[80] J. Aranda and S. Easterbrook, "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*, New York, NY, USA, 2005, pp. 346–355. doi: 10.1145/1081706.1081761.

[81] M. Jorgensen and S. Grimstad, "Software Development Estimation Biases: The Role of Interdependence," *IEEE Transactions on Software Engineering*, vol. 38, no. 3, pp. 677–693, May 2012, doi: 10.1109/TSE.2011.40.

[82] M. Jørgensen and T. M. Gruschke, "The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments," *IEEE Transactions on Software Engineering*, vol. 35, no. 3, pp. 368–383, May 2009, doi: 10.1109/TSE.2009.2.

[83] D. Yang, Q. Wang, M. Li, Y. Yang, K. Ye, and J. Du, "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*, Kaiserslautern, Germany, 2008, p. 253. doi: 10.1145/1414004.1414045.

[84] J. Rahikkala, S. Hyrynsalmi, and V. Leppänen, "Accounting Testing in Software Cost Estimation: A Case Study of the Current Practice and Impacts," in *14th Symposium on Programming Languages and Software Tools*, Tampere, Finland, 2015, p. 15.
[85] A. L. Lederer and J. Prasad, "Causes of inaccurate software development cost estimates," *Journal of Systems and Software*, vol. 31, no. 2, pp. 125–134, Nov. 1995, doi: 10.1016/0164-1212(94)00092-2.
[86] M. Usman, R. Britto, L.-O. Damm, and J. Börstler, "Effort estimation in large-scale software development: An industrial case study," *Information and Software Technology*, vol. 99, pp. 21–40, Jul. 2018, doi: 10.1016/j.infsof.2018.02.009.
[87] A. Lederer and R. Mirani, "Information System Cost Estimating: A Management Perspective," *Management Information Systems Quarterly*, vol. 14, no. 2, pp. 159–176, Dec. 1990.
[88] O. Matos, L. Fortaleza, T. Conte, and E. Mendes, "Realising web effort estimation," in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, Porto de Galinhas, 2013, pp. 12–23. doi: 10.1145/2460999.2461002.
[89] S. Grimstad, M. Jorgensen, and K. Molokken-Ostvold, "The clients' impact on effort estimation accuracy in software development projects," in *11th IEEE International Software Metrics Symposium (METRICS'05)*, Como, Italy, Sep. 2005, p. 10 pp. – 10. doi: 10.1109/METRICS.2005.30.
[90] M. He, H. Zhang, Y. Yang, Q. Wang, and M. Li, "Understanding the Influential Factors to Development Effort in Chinese Software Industry," in *Product-Focused Software Process Improvement*, Berlin, Heidelberg, 2010, pp. 306–320. doi: 10.1007/978-3-642-13792-1_24.
[91] J. Huang, H. Sun, and Y. Li, "An empirical study of the impact of project factors on software economics," in *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Dec. 2015, pp. 43–47. doi: 10.1109/IEEM.2015.7385605.
[92] M. Jørgensen and S. Grimstad, "Avoiding Irrelevant and Misleading Information When Estimating Development Effort," *IEEE Software*, vol. 25, no. 3, pp. 78–83, May 2008, doi: 10.1109/MS.2008.57.
[93] M. Jørgensen and D. I. K. Sjøberg, "The impact of customer expectation on software development effort estimates," *International Journal of Project Management*, vol. 22, no. 4, pp. 317–325, May 2004, doi: 10.1016/S0263-7863(03)00085-1.
[94] M. Conoscenti, V. Besner, A. Vetrò, and D. M. Fernández, "Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development," *Journal of Systems and Software*, vol. 156, pp. 126–135, Oct. 2019, doi: 10.1016/j.jss.2019.06.075.
[95] K. M. Furulund and K. Molkken-stvold, "Increasing Software Effort Estimation Accuracy Using Experience Data, Estimation Models and Checklists," in *Seventh International Conference on Quality Software (QSIC 2007)*, Portland, OR, USA, Oct. 2007, pp. 342–347. doi: 10.1109/QSIC.2007.4385518.
[96] M. Jorgensen and K. Molokken-Ostvold, "Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method," *IEEE Transactions on Software Engineering*, vol. 30, no. 12, pp. 993–1007, Dec. 2004, doi: 10.1109/TSE.2004.103.
[97] A. Altaleb, M. Altherwi, and A. Gravell, "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*, 2020. Accessed: Nov. 26, 2020. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3388176.3388212
[98] A. Zarour and S. Zein, "Software Development Estimation Techniques in Industrial Contexts: An Exploratory Multiple Case-Study," *International Journal of Technology in Education and Science*, vol. 3, no. 2, pp. 72–84, 2019.
[99] T. Arnuphaptrairong, "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, 2018, vol. 2, p. 6.
[100] L. Layman, N. Nagappan, S. Guckenheimer, J. Beehler, and A. Begel, "Mining software effort data: preliminary analysis of visual studio team system data," in *Proceedings of the 2008 international working conference on Mining software repositories*, New York, NY, USA, May 2008, pp. 43–46. doi: 10.1145/1370750.1370762.
[101] A. H. Zapata and M. R. V. Chaudron, "An empirical study into the accuracy of it estimations and its influencing factors," *Int. J. Soft. Eng. Knowl. Eng.*, vol. 23, no. 04, pp. 409–432, May 2013, doi: 10.1142/S0218194013400081.
[102] S. Keaveney and K. Conboy, "Cost estimation in agile development projects," in *ECIS 2006 Proceedings*, Göteborg, Sweden, 2006, p. 16.
[103] M. Usman, J. Börstler, and K. Petersen, "An Effort Estimation Taxonomy for Agile Software Development," *Int. J. Soft. Eng. Knowl. Eng.*, vol. 27, no. 04, pp. 641–674, May 2017, doi: 10.1142/S0218194017500243.
[104] K. Molokken-Ostvold and K. M. Furulund, "The Relationship between Customer Collaboration and Software Project Overruns," in *Agile 2007 (AGILE 2007)*, Aug. 2007, pp. 72–83. doi: 10.1109/AGILE.2007.57.
[105] M. Jørgensen and K. Moløkken, "Combination of software development effort prediction intervals: why, when and how?," in *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, Ischia, Italy, Jul. 2002, pp. 425–428. doi: 10.1145/568760.568833.
[106] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2086–2095, Sep. 2012, doi: 10.1016/j.jss.2012.04.005.
[107] T. J. Gandomani, H. Faraji, and M. Radnejad, "Planning Poker in cost estimation in Agile methods: Averaging Vs. Consensus," in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, Tehran, Iran, Feb. 2019, pp. 066–071. doi: 10.1109/KBEI.2019.8734960.
[108] N. C. Haugen, "An empirical study of using planning poker for user story estimation," in *AGILE 2006 (AGILE'06)*, Minneapolis, MN, USA, Jul. 2006, p. 9 pp. – 34. doi: 10.1109/AGILE.2006.16.
[109] K. Moløkken-Østvold and M. Jørgensen, "Group Processes in Software Effort Estimation," *Empirical Software Engineering*, vol. 9, no. 4, pp. 315–334, Dec. 2004, doi: 10.1023/B:EMSE.0000039882.39206.5a.

[110] K. Moløkken-Østvold, N. C. Haugen, and H. C. Benestad, "Using planning poker for combining expert estimates in software projects," *Journal of Systems and Software*, vol. 81, no. 12, pp. 2106–2117, Dec. 2008, doi: 10.1016/j.jss.2008.03.058.

[111] M. Usman, K. Petersen, J. Börstler, and P. Santos Neto, "Developing and using checklists to improve software effort estimation: A multi-case study," *Journal of Systems and Software*, vol. 146, pp. 286–309, Dec. 2018, doi: 10.1016/j.jss.2018.09.054.

[112] A. Magazinius and R. B. Svensson, "Effects of Feature Complexity on Software Effort Estimates – An Exploratory Study," in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, Aug. 2014, pp. 301–304. doi: 10.1109/SEAA.2014.69.

[113] O. Morgenshtern, T. Raz, and D. Dvir, "Factors affecting duration and effort estimation errors in software development projects," *Information and Software Technology*, vol. 49, no. 8, pp. 827–837, Aug. 2007, doi: 10.1016/j.infsof.2006.09.006.

[114] M. Lee, M. Rothenberger, and K. Peffers, "Identifying Effort Estimation Factors for Corrective Maintenance in Object-Oriented Systems," in *AMCIS 2011 Proceedings*, Aug. 2011, p. 186. [Online]. Available: https://aisel.aisnet.org/amcis2011_submissions/186

[115] T. Silva-de-Souza and G. H. Travassos, "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*, New York, NY, USA, Sep. 2017, pp. 1–10. doi: 10.1145/3128473.3128480.

[116] G. H. Subramanian, P. C. Pendharkar, and M. Wallace, "An empirical study of the effect of complexity, platform, and program type on software development effort of business applications," *Empir Software Eng*, vol. 11, no. 4, pp. 541–553, Dec. 2006, doi: 10.1007/s10664-006-9023-3.

[117] A. Altaleb, M. Altherwi, and A. Gravell, "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)*, Oxford, UK, Feb. 2020, pp. 291–296. doi: 10.1109/ICITM48982.2020.9080362.

[118] R. Britto, E. Mendes, and J. Börstler, "An Empirical Investigation on Effort Estimation in Agile Global Software Development," in *2015 IEEE 10th International Conference on Global Software Engineering*, Jul. 2015, pp. 38–45. doi: 10.1109/ICGSE.2015.10.

[119] D. Basten and W. Mellis, "A Current Assessment of Software Development Effort Estimation," in *2011 International Symposium on Empirical Software Engineering and Measurement*, Banff, AB, Canada, Sep. 2011, pp. 235–244. doi: 10.1109/ESEM.2011.32.

[120] M. Jørgensen and S. Grimstad, "The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment," *IEEE Transactions on Software Engineering*, vol. 37, no. 5, pp. 695–707, Sep. 2011, doi: 10.1109/TSE.2010.78.

[121] S. Grimstad and M. Jorgensen, "The Impact of Irrelevant Information on Estimates of Software Development Effort," in *2007 Australian Software Engineering Conference (ASWEC'07)*, Apr. 2007, pp. 359–368. doi: 10.1109/ASWEC.2007.48.

[122] M. Jorgensen and G. J. Carelius, "An Empirical Study of Software Project Bidding," *IEEE Trans. Softw. Eng.*, vol. 30, no. 12, pp. 953–969, Dec. 2004, doi: 10.1109/TSE.2004.92.

[123] J. Rahikkala, S. Hyrynsalmi, V. Leppänen, and I. Porres, "The Role of Organisational Phenomena in Software Cost Estimation: A Case Study of Supporting and Hindering Factors," *e-Informatica Software Engineering Journal*, vol. 12, no. 1, pp. 167–198, 2018, doi: 10.5277/e-Inf180107.

[124] A. Magazinovic and J. Pernstål, "Any other cost estimation inhibitors?," in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM '08*, Kaiserslautern, Germany, 2008, p. 233. doi: 10.1145/1414004.1414042.

[125] J. Davis, "Investigation of predictors of failures and debugging effort for large MIS," *Information and Software Technology*, vol. 31, no. 4, pp. 170–174, May 1989, doi: 10.1016/0950-5849(89)90034-7.

[126] M. Jørgensen and D. I. K. Sjøberg, "Impact of effort estimates on software project work," *Information and Software Technology*, vol. 43, p. 10, 2001.

[127] A. L. Lederer and J. Prasad, "A causal model for software cost estimating error," *IEEE Transactions on Software Engineering*, vol. 24, no. 2, pp. 137–148, Feb. 1998, doi: 10.1109/32.666827.

[128] A. Altaleb and A. Gravell, "An Empirical Investigation of Effort Estimation in Mobile Apps Using Agile Development Process," *Journal of Software*, vol. 14, no. 8, pp. 356–369, 2019.

[129] R. Lagerström, L. M. von Würtemberg, H. Holm, and O. Luczak, "Identifying factors affecting software development cost and productivity," *Software Qual J*, vol. 20, no. 2, pp. 395–417, Jun. 2012, doi: 10.1007/s11219-011-9137-8.

[130] R. L. Glass, J. Rost, and M. S. Matook, "Lying on Software Projects," *IEEE Software*, vol. 25, no. 6, pp. 90–95, Nov. 2008, doi: 10.1109/MS.2008.150.

[131] M. Jørgensen, B. Faugli, and T. Gruschke, "Characteristics of software engineers with optimistic predictions," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1472–1482, Sep. 2007, doi: 10.1016/j.jss.2006.09.047.

[132] H. Karna and S. Gotovac, "Estimators characteristics and effort estimation of software projects," in *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, Vienna, Austria, Aug. 2014, pp. 26–35.

[133] A. L. Lederer and J. Prasad, "The validation of a political model of information systems development cost estimating," *ACM SIGCPR Computer Personnel*, vol. 13, no. 2, pp. 47–57, Aug. 1991, doi: 10.1145/122393.122398.

[134] S. Grimstad and M. Jørgensen, "Preliminary study of sequence effects in judgment-based software development work-effort estimation," *IET Software*, vol. 3, no. 5, pp. 435–441, Oct. 2009, doi: 10.1049/iet-sen.2008.0110.

[135] M. Jørgensen, "Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays," *Journal of Systems and Software*, vol. 117, pp. 274–281, Jul. 2016, doi: 10.1016/j.jss.2016.03.048.

[136] M. Jørgensen and T. Halkjelsvik, "Sequence effects in the estimation of software development effort," *Journal of Systems and Software*, vol. 159, p. 110448, Jan. 2020, doi: 10.1016/j.jss.2019.110448.
[137] M. Jørgensen, "Relative Estimation of Software Development Effort: It Matters with What and How You Compare," *IEEE Software*, vol. 30, no. 2, pp. 74–79, Mar. 2013, doi: 10.1109/MS.2012.70.
[138] S. Vijayakumar, "Use of historical data In software cost estimation," *Computing Control Engineering Journal*, vol. 8, no. 3, pp. 113–119, Jun. 1997, doi: 10.1049/cce:19970303.
[139] J. Hill, L. C. Thomas, and D. E. Allen, "Experts' estimates of task durations in software development projects," *International Journal of Project Management*, vol. 18, no. 1, pp. 13–21, Feb. 2000, doi: 10.1016/S0263-7863(98)00062-3.
[140] S. Halstead, R. Ortiz, M. Córdova, and M. Seguí, "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*, Berlin, Heidelberg, Jun. 2012, pp. 248–259. doi: 10.1007/978-3-642-31063-8_19.
[141] M. Jorgensen, G. R. Bergersen, and K. Liestol, "Relations Between Effort Estimates, Skill Indicators, and Measured Programming Skill," *IEEE Transactions on Software Engineering*, pp. 1–1, 2020, doi: 10.1109/TSE.2020.2973638.
[142] M. Jørgensen and T. Halkjelsvik, "The effects of request formats on judgment-based effort estimation," *Journal of Systems and Software*, vol. 83, no. 1, pp. 29–36, Jan. 2010, doi: 10.1016/j.jss.2009.03.076.
[143] T. Halkjelsvik and M. Jorgensen, "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*, vol. 25, no. 2, pp. 314–323, 2011, doi: 10.1002/acp.1693.
[144] V. Lenarduzzi, "Could social factors influence the effort software estimation?," in *Proceedings of the 7th International Workshop on Social Software Engineering*, New York, NY, USA, Sep. 2015, pp. 21–24. doi: 10.1145/2804381.2804385.
[145] M. R. Lind and J. M. Sulek, "Undersizing software systems: third versus fourth generation software development," *Eur. J. Inf. Syst.*, vol. 7, no. 4, pp. 261–268, Dec. 1998, doi: 10.1038/sj.ejis.3000308.
[146] M. Jørgensen, "The effect of the time unit on software development effort estimates," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Kathmandu, Nepal, Dec. 2015, pp. 1–5. doi: 10.1109/SKIMA.2015.7399992.
[147] K. Molokken-Ostvold and M. Jorgensen, "A comparison of software project overruns - flexible versus sequential development models," *IEEE Transactions on Software Engineering*, vol. 31, no. 9, pp. 754–766, Sep. 2005, doi: 10.1109/TSE.2005.96.
[148] S. Koch and G. Turk, "Human Resource Related Problems in Agile and Traditional Software Project Process Models," *Int. J. Inf. Technol. Proj. Manag.*, vol. 2, no. 2, pp. 1–13, Apr. 2011, doi: 10.4018/jitpm.2011040101.
[149] M. Brown, H. Dirska, M. Pelosi, and M. Assadullah, "Agile Method Software Development Estimation Biases," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 10, 2013.
[150] O. Shmueli, N. Pliskin, and L. Fink, "Can the outside-view approach improve planning decisions in software development projects?," *Information Systems Journal*, vol. 26, no. 4, pp. 395–418, 2016, doi: https://doi.org/10.1111/isj.12091.
[151] U. Passing and M. Shepperd, "An Experiment on Software Project Size and Effort Estimation," in *Proceedings of the 2003 International Symposium on Empirical Software Engineering*, Washington, DC, USA, 2003, pp. 120-. Accessed: Dec. 19, 2019. [Online]. Available: http://dl.acm.org/citation.cfm?id=942801.943632
[152] R. Valerdi, "Cognitive Limits of Software Cost Estimation," in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, Sep. 2007, pp. 117–125. doi: 10.1109/ESEM.2007.85.
[153] M. Jørgensen, "Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation," *Information and Software Technology*, vol. 53, no. 12, pp. 1382–1390, Dec. 2011, doi: 10.1016/j.infsof.2011.07.001.
[154] K. Moløkken and M. Jørgensen, "Expert Estimation of Web-Development Projects: Are Software Professionals in Technical Roles More Optimistic Than Those in Non-Technical Roles?," *Empirical Software Engineering*, vol. 10, no. 1, pp. 7–30, Jan. 2005, doi: 10.1023/B:EMSE.0000048321.46871.2e.
[155] M. Jørgensen and E. Løhre, "First Impressions in Software Development Effort Estimation: Easy to Create and Difficult to Neutralize," 2012, pp. 216–222.
[156] M. Jørgensen, "Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates," *Information and Software Technology*, vol. 52, no. 5, pp. 506–516, May 2010, doi: 10.1016/j.infsof.2009.12.002.
[157] L. Gren, R. B. Svensson, and M. Unterkalmsteiner, "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)*, May 2017, pp. 56–61. doi: 10.1109/CHASE.2017.10.
[158] M. Atas, S. Reiterer, A. Felfernig, T. N. T. Tran, and M. Stettinger, "Polarization Effects in Group Decisions," in *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, New York, NY, USA, Jul. 2018, pp. 305–310. doi: 10.1145/3213586.3225242.
[159] M. Jørgensen, "Selection of strategies in judgment-based effort estimation - ScienceDirect," *Journal of Systems and Software*, vol. 83, no. 6, pp. 1039–1050, 2010.
[160] N. Tripathi, P. Seppänen, M. Oivo, J. Similä, and K. Liukkunen, "The Effect of Competitor Interaction on Startup's Product Development," in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Vienna, Austria, Aug. 2017, pp. 125–132. doi: 10.1109/SEAA.2017.34.
[161] J. McDonald, "The Impact of Project Planning Team Experience on Software Project Cost Estimates," *Empir Software Eng*, vol. 10, no. 2, pp. 219–234, Apr. 2005, doi: 10.1007/s10664-004-6192-9.

[162] G. Subramanian, P. C. Pendharkar, and D. Pai, "An Examination of Determinants of Software Testing and Project Management Effort," *Journal of Computer Information Systems*, vol. 57, no. 2, pp. 123–129, 2017.

[163] M. Jørgensen, "Individual Differences in How Much People Are Affected by Irrelevant and Misleading Information," presented at the Second European Conference on Cognitive Science, 2007, pp. 347–352.

[164] M. Jørgensen, "The Ignorance of Confidence Levels in Minimum-Maximum Software Development Effort Intervals," *LNSE*, vol. 2, no. 4, pp. 327–330, 2014, doi: 10.7763/LNSE.2014.V2.144.

[165] R. Tamrakar and M. Jørgensen, "Does the use of Fibonacci numbers in planning poker affect effort estimates?," in *16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, Ciudad Real, May 2012, pp. 228–232. doi: 10.1049/ic.2012.0030.

[166] M. Jørgensen and K. Moløkken, "Eliminating Over-Confidence in Software Development Effort Estimates," in *Product Focused Software Process Improvement*, Berlin, Heidelberg, 2004, pp. 174–184. doi: 10.1007/978-3-540-24659-6_13.

[167] M. Jørgensen, K. H. Teigen, and K. Moløkken, "Better sure than safe? Over-confidence in judgement based software development effort prediction intervals," *Journal of Systems and Software*, vol. 70, no. 1, pp. 79–93, Feb. 2004, doi: 10.1016/S0164-1212(02)00160-7.

[168] B. Tanveer, L. Guzmán, and U. M. Engel, "Effort estimation in agile software development: Case study and improvement framework," *Journal of Software: Evolution and Process*, vol. 29, no. 11, p. e1862, 2017, doi: https://doi.org/10.1002/smr.1862.

[169] R. M. Henry, G. E. McCray, R. L. Purvis, and T. L. Roberts, "Exploiting organizational knowledge in developing IS project cost and schedule estimates: An empirical study," *Information & Management*, vol. 44, no. 6, pp. 598–612, Sep. 2007, doi: 10.1016/j.im.2007.06.002.

[170] S. Suliman and G. Kadoda, "Factors that influence software project cost and schedule estimation," presented at the Sudan Conference on Computer Science and Information Technology, Elnihood, Sudan, 2017. Accessed: Feb. 10, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/8293053

[171] E. Mendes, N. Mosley, and S. Counsell, "Investigating Web size metrics for early Web cost estimation," *Journal of Systems and Software*, vol. 77, no. 2, pp. 157–172, Aug. 2005, doi: 10.1016/j.jss.2004.08.034.

[172] A. L. Lederer and J. Prasad, "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*, Nashville, Tennessee, USA, Apr. 1995, pp. 50–59. doi: 10.1145/212490.212504.

[173] J. Rahikkala, V. Leppänen, J. Ruohonen, and J. Holvitie, "Top management support in software cost estimation: A study of attitudes and practice in Finland," *International Journal of Managing Projects in Business*, vol. 8, no. 3, pp. 513–532, Jan. 2015, doi: 10.1108/IJMPB-11-2014-0076.

[174] R. Rozalina and Z. Mansor, "Validated Software Cost Estimation Factors for Government Projects using Rasch Measurement Model," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 5, Art. no. 5, Oct. 2018, doi: 10.18517/ijaseit.8.5.6386.

[175] M. A. Ramessur and S. D. Nagowah, "Factors Affecting Sprint Effort Estimation," in *Advanced Computing and Intelligent Engineering*, Singapore, 2020, pp. 507–518. doi: 10.1007/978-981-15-1483-8_43.

[176] M. Jørgensen, "The Use of Precision of Software Development Effort Estimates to Communicate Uncertainty," in *Software Quality. The Future of Systems and Software Development*, Cham, 2016, pp. 156–168. doi: 10.1007/978-3-319-27033-3_11.

[177] S. Grimstad and M. Jørgensen, "Inconsistency of expert judgment-based estimates of software development effort," *Journal of Systems and Software*, vol. 80, no. 11, pp. 1770–1777, Nov. 2007, doi: 10.1016/j.jss.2007.03.001.

[178] S. S. Vicinanza, T. Mukhopadhyay, and M. J. Prietula, "Software-Effort Estimation: An Exploratory Study of Expert Performance," *Info. Sys. Research*, vol. 2, no. 4, pp. 243–262, Dec. 1991, doi: 10.1287/isre.2.4.243.

[179] T. M. Gruschke and M. Jørgensen, "The role of outcome feedback in improving the uncertainty assessment of software development effort estimates," *ACM Trans. Softw. Eng. Methodol.*, vol. 17, no. 4, p. 20:1-20:35, Aug. 2008, doi: 10.1145/13487689.13487693.

[180] M. Host and C. Wohlin, "An experimental study of individual subjective effort estimations and combinations of the estimates," in *Proceedings of the 20th International Conference on Software Engineering*, Apr. 1998, pp. 332–339. doi: 10.1109/ICSE.1998.671386.

[181] L. Cao, "Estimating Agile Software Project Effort: An Empirical Study," presented at the Americas Conference on Information Systems, 2008, p. 11.

[182] M. Jørgensen, "The influence of selection bias on effort overruns in software development projects," *Information and Software Technology*, vol. 55, no. 9, pp. 1640–1650, Sep. 2013, doi: 10.1016/j.infsof.2013.03.001.

[183] S. Grapenthin, M. Book, T. Richter, and V. Gruhn, "Supporting Feature Estimation with Risk and Effort Annotations," in *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Limassol, Cyprus, Aug. 2016, pp. 17–24. doi: 10.1109/SEAA.2016.24.

[184] D. T. Martínez C. Branco, E. C. Cunha de Oliveira, L. Galvão, R. Prikladnicki, and T. Conte, "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*, Portugal, 2015, pp. 102–113. doi: 10.5220/0005373001020113.

[185] F. Bergeron and J.-Y. St-Arnaud, "Estimation of information systems development efforts: A pilot study," *Information & Management*, vol. 22, no. 4, pp. 239–254, Apr. 1992, doi: 10.1016/0378-7206(92)90026-C.

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

[187] M. C. Ohlsson, C. Wohlin, and B. Regnell, "A project effort estimation study," *Information and Software Technology*, vol. 40, no. 14, pp. 831–839, Dec. 1998, doi: 10.1016/S0950-5849(98)00097-4.

[188] H. H. Arifin, J. Daengdej, and N. T. Khanh, "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)*, Tokyo, Japan, Mar. 2017, pp. 35–40. doi: 10.1109/IWESEP.2017.21.

[189] A. Nugroho and C. F. J. Lange, "On the Relation between Class-Count and Modeling Effort," in *Models in Software Engineering*, Berlin, Heidelberg, 2008, pp. 93–104. doi: 10.1007/978-3-540-69073-3_11.

[190] N. Benschop, C. A. R. Hilhorst, A. L. P. Nuijten, and M. Keil, "Detection of early warning signals for overruns in IS projects: linguistic analysis of business case language," *European Journal of Information Systems*, vol. 29, no. 2, pp. 190–202, Mar. 2020, doi: 10.1080/0960085X.2020.1742587.

[191] A. R. Gray, S. G. MacDonell, and M. J. Shepperd, "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)*, Nov. 1999, pp. 216–227. doi: 10.1109/METRIC.1999.809743.

[192] M. Jorgensen, "Realism in assessment of effort estimation uncertainty: it matters how you ask," *IEEE Transactions on Software Engineering*, vol. 30, no. 4, pp. 209–217, Apr. 2004, doi: 10.1109/TSE.2004.1274041.

[193] M. Jørgensen, "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)*, Dec. 2018, pp. 19–24. doi: 10.1109/IWESEP.2018.00012.

[194] M. Jørgensen and K.-H. Teigen, "Uncertainty Intervals Versus Interval Uncertainty: an Alternative Method for Eliciting Effort Prediction Intervals in Software Development Projects," presented at the International Conference on Project Management, Singapore, 2002, pp. 343–352.

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

[196] G. Boetticher and N. Lokhandwala, "Assessing the Reliability of a Human Estimator," presented at the Third International Workshop on Predictor Models in Software Engineering (PROMISE'07), USA, 2007. Accessed: Jan. 07, 2020. [Online]. Available: https://dl.acm.org/doi/pdf/10.1109/PROMISE.2007.2

[197] S. Bukhari and A. A. Malik, "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*, Dec. 2012, pp. 41–45. doi: 10.1109/FIT.2012.16.

[198] A. Javed, U. M. A., and R. Aziz-Ur, "Factors Affecting Software Cost Estimation in Developing Countries," *International Journal of Information Technology and Computer Science*, vol. 5, pp. 54–59, 2013.

[199] L. M. Taff, J. W. Borchering, and W. R. Hudgins, "Estimeetings: development estimates and a front-end process for a large project," *IEEE Transactions on Software Engineering*, vol. 17, no. 8, pp. 839–849, Aug. 1991, doi: 10.1109/32.83918.

[200] L. Bratthall, E. Arisholm, and M. Jørgensen, "Program Understanding Behavior During Estimation of Enhancement Effort on Small Java Programs | SpringerLink," presented at the Product Focused Software Process Improvement, 2001. Accessed: Feb. 11, 2021. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-44813-6_30

[201] M. Jørgensen, "Fallacies and Biases when Adding Effort Estimates," in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, Verona, Italy, Aug. 2014, pp. 277–284. doi: 10.1109/SEAA.2014.16.

[202] M. Jorgensen and S. Grimstad, "Over-optimism in software development projects: 'the winner's curse,'" in *15th International Conference on Electronics, Communications and Computers (CONIELECOMP'05)*, Feb. 2005, pp. 280–285. doi: 10.1109/CONIEL.2005.58.

[203] T. Little, "Schedule estimation and uncertainty surrounding the cone of uncertainty," *IEEE Software*, vol. 23, no. 3, pp. 48–54, May 2006, doi: 10.1109/MS.2006.82.

[204] M. Jørgensen, "Communication of Software Cost Estimates," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, New York, NY, USA, 2014, p. 28:1-28:5. doi: 10.1145/2601248.2601262.

[205] M. Jørgensen, "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, 2007.

[206] B. W. Boehm, "Software Engineering Economics," *IEEE Transactions on Software Engineering*, vol. SE-10, no. 1, pp. 4–21, Jan. 1984, doi: 10.1109/TSE.1984.5010193.

[207] K.-J. Stol and B. Fitzgerald, "The ABC of Software Engineering Research," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 3, p. 11:1-11:51, Sep. 2018, doi: 10.1145/3241743.

[208] D. Kahneman, O. Sibony, and C. R. Sunstein, "Debiasing and decision hygiene," in *Noise: A Flaw in Human Judgment*, New York: Little, Brown Spark, 2021, pp. 203–209.

[209] T. Halkjelsvik and M. Jørgensen, *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*. Cham, Switzerland: Springer International Publishing, 2018. doi: 10.1007/978-3-319-74953-2.

[210] D. Kahneman, O. Sibony, and C. R. Sunstein, "Introduction: two kinds of error," in *Noise: A Flaw in Human Judgment*, New York: Little, Brown Spark, 2021, pp. 9–15.

[211] D. Kahneman, O. Sibony, and C. R. Sunstein, "Your mind is a measuring instrument," in *Noise: A Flaw in Human Judgment*, New York: Little, Brown Spark, 2021, pp. 39–41.

[212] A. Ampatzoglou, S. Bibi, P. Avgeriou, M. Verbeek, and A. Chatzigeorgiou, "Identifying, categorizing and mitigating threats to validity in software engineering secondary studies," *Information and Software Technology*, vol. 106, pp. 201–230, Feb. 2019, doi: 10.1016/j.infsof.2018.10.006.

[213] M. Jorgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 33–53, Jan. 2007, doi: 10.1109/TSE.2007.256943.

[214] V. Ivanov, A. Rogers, G. Succi, J. Yi, and V. Zorin, "What Do Software Engineers Care About? Gaps Between Research and Practice," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, New York, NY, USA, 2017, pp. 890–895. doi: 10.1145/3106237.3117778.

[215] M. Jorgensen, "Practical guidelines for expert-judgment-based software effort estimation," *IEEE Software*, vol. 22, no. 3, pp. 57–63, May 2005, doi: 10.1109/MS.2005.73.
[216] S. McConnell, *Software Estimation: Demystifying the Black Art*, 1st ed. Redomnd, Washington, USA: Microsoft Press, 2006.
[217] A. Magazinius and R. Feldt, "Exploring the human and organizational aspects of software cost estimation," in *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, New York, NY, USA, May 2010, pp. 92–95. doi: 10.1145/1833310.1833325.
[218] P. Tell *et al.*, "What are Hybrid Development Methods Made Of? An Evidence-Based Characterization," in *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, May 2019, pp. 105–114. doi: 10.1109/ICSSP.2019.00022.
[219] Z. Masood, R. Hoda, and K. Blincoe, "Real World Scrum A Grounded Theory of Variations in Practice," *IEEE Transactions on Software Engineering*, pp. 1–1, 2020, doi: 10.1109/TSE.2020.3025317.
[220] J. Corbin and A. Strauss, *Basics of qualitative research - Techniques and procedures for developing grounded theory*, 4th ed. Sage Publications, 2014.
[221] S. McConnell, "How to defend an unpopular schedule [software development projects]," *IEEE Software*, vol. 13, no. 3, pp. 120–119, May 1996, doi: 10.1109/52.493033.
[222] T. deMarco and T. Lister, *Waltzing with bears - Managing risk on software projects*. Dorset House Publishing, 2003.
[223] E. Lim, N. Taksande, and C. Seaman, "A Balancing Act: What Software Practitioners Have to Say about Technical Debt," *IEEE Software*, vol. 29, no. 6, pp. 22–27, Nov. 2012, doi: 10.1109/MS.2012.130.
[224] Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," *J. Syst. Softw.*, vol. 101, no. C, pp. 193–220, Mar. 2015, doi: 10.1016/j.jss.2014.12.027.
[225] C. Becker, D. Walker, and C. McCord, "Intertemporal Choice: Decision Making and Time in Software Engineering," in *2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, May 2017, pp. 23–29. doi: 10.1109/CHASE.2017.6.
[226] S. Frederick, G. Loewenstein, and T. O'Donoghue, "Time Discounting and Time Preference: A Critical Review," *Journal of Economic Literature*, vol. 40, no. 2, pp. 351–401, 2002.
[227] O. Chernikova, N. Heitzmann, M. Stadler, D. Holzberger, T. Seidel, and F. Fischer, "Simulation-Based Learning in Higher Education: A Meta-Analysis," *Review of Educational Research*, vol. 90, no. 4, pp. 499–541, Aug. 2020, doi: 10.3102/0034654320933544.
[228] V. Veloso Gouveia *et al.*, "A Short Version of the Big Five Inventory (BFI-20): Evidence on Construct Validity," *RIP/IJP*, vol. 55, no. 1, p. e1312, Apr. 2021, doi: 10.30849/ripijp.v55i1.1312.
[229] L. Pasquali and V. V. Gouveia, "Escala de assertividade Rathus - Ras:: Adaptação brasileira," *Psic.: Teor. e Pesq.*, vol. 6, no. 3, pp. 233–249, Aug. 2012.
[230] H. D. Tehrani and S. Yamini, "Personality traits and conflict resolution styles: A meta-analysis," *Personality and Individual Differences*, vol. 157, p. 109794, Apr. 2020, doi: 10.1016/j.paid.2019.109794.
[231] H. Bless and A. M. Burger, "Assimilation and contrast in social priming," *Current Opinion in Psychology*, vol. 12, pp. 26–31, 2016, doi: 10.1016/j.copsyc.2016.04.018.
[232] J. Lammers, D. Dubois, D. D. Rucker, and A. D. Galinsky, "Power gets the job: Priming power improves interview outcomes," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 776–779, Jul. 2013, doi: 10.1016/j.jesp.2013.02.008.
[233] C. A. Furia, "What good is bayesian data analysis for software engineering?," in *Proceedings of the 39th International Conference on Software Engineering Companion*, Buenos Aires, Argentina, Maio 2017, pp. 374–376. doi: 10.1109/ICSE-C.2017.92.
[234] C. A. Furia, R. Feldt, and R. Torkar, "Bayesian Data Analysis in Empirical Software Engineering Research," *IEEE Transactions on Software Engineering*, vol. 47, no. 9, pp. 1786–1810, Sep. 2021, doi: 10.1109/TSE.2019.2935974.
[235] H. Erdogmus, "Bayesian Hypothesis Testing Illustrated: An Introduction for Software Engineering Researchers," *ACM Comput. Surv.*, Abril 2022, doi: 10.1145/3533383.
[236] R. Torkar *et al.*, "A Method to Assess and Argue for Practical Significance in Software Engineering," *IEEE Transactions on Software Engineering*, vol. 48, no. 6, pp. 2053–2065, Jun. 2022, doi: 10.1109/TSE.2020.3048991.
[237] S. Baltes and P. Ralph, "Sampling in software engineering research: a critical review and guidelines," *Empir Software Eng*, vol. 27, no. 4, p. 94, Apr. 2022, doi: 10.1007/s10664-021-10072-8.
[238] P. Matsubara, I. Steinmacher, J. Maldonado, B. Gadelha, and T. Conte, "Trust yourself! Or maybe not: factors related to overconfidence and uncertainty assessments of software effort estimates," in *Brazilian Symposium on Software Engineering*, New York, NY, USA, Setembro 2021, pp. 452–461. doi: 10.1145/3474624.3474643.
[239] R. H. Thaler and C. R. Sunstein, *Nudge: The Final Edition*, Final Edition. New York: Penguin Books, 2021.
[240] M. Cohn, *Agile Estimating and Planning*, 1ª edição. Pearson, 2005.

APPENDIX A – REMAINING QUESTIONS FROM THE SLM

This appendix contains the answers to the remaining questions from the SLM.

Regarding the project variables investigated in the primary studies, we extracted the metrics that authors reported as within their studies' scope. Figure 1 shows the results we obtained, making evident that most of the studies focus on effort estimation.

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.

Our sample includes papers published between 1989 and 2020. The past two decades have been very fruitful regarding research about factors affecting estimates, as shown in Figure 2, 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.

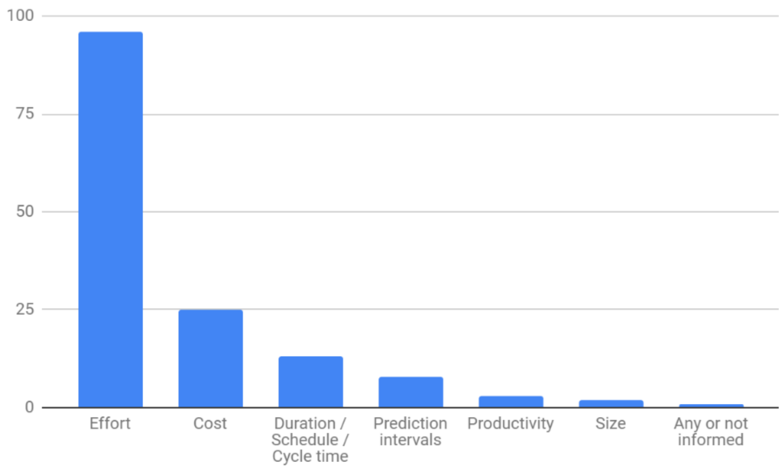


Figure 1 – Variables investigated.

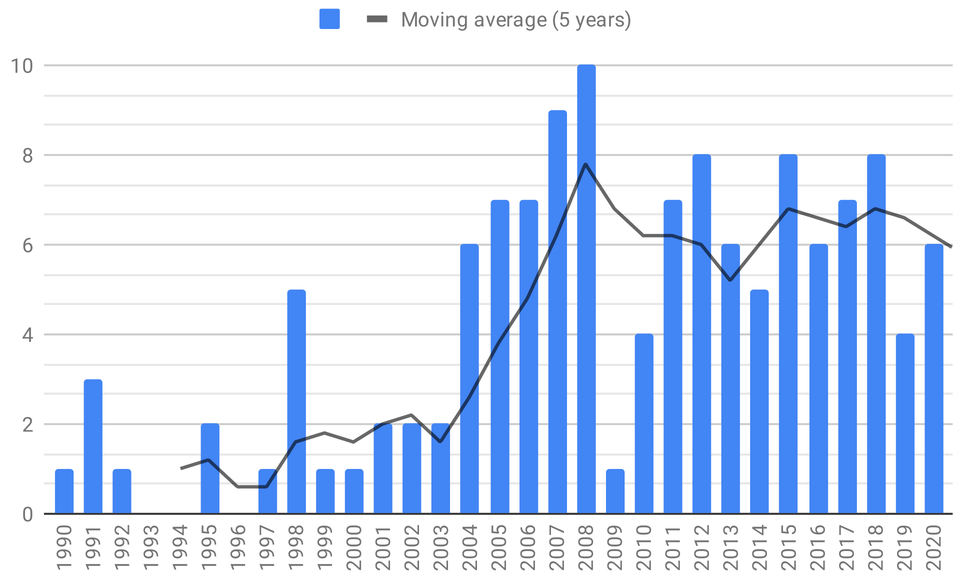


Figure 2 – Research papers per year.

Table 1 shows all the venues concentrating three or more studies about factors affecting estimates. In total, we represent 65 papers in Table 1.

Table 1 – 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 papers.

To answer SQ 1.5, we classified the studies considering the taxonomies proposed by Storey et al. [78], which is focused on human factors of software engineering, identifying four research strategies: respondents, lab, field, and data, as we show in Table 2. Each paper can report more than one study and, accordingly, could be associated with more than one research strategy.

Table 2 – 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 [78]. 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 3, we show the use of the 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.

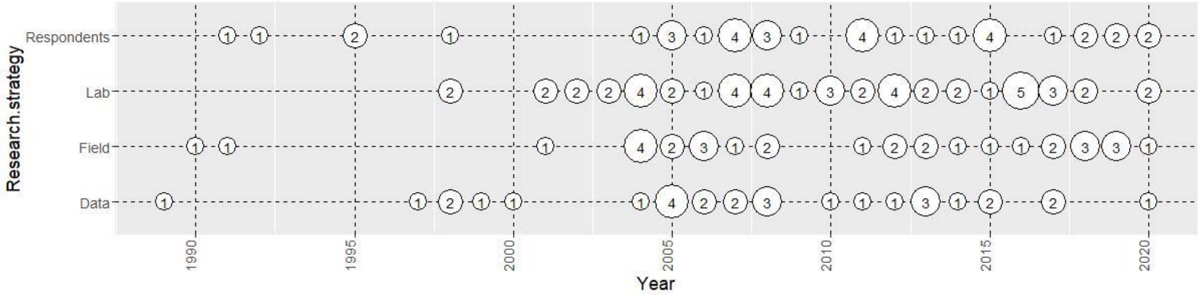


Figure 3 - Research strategies throughout the years.

APPENDIX B – THE BOOKLET

This appendix presents the booklet describing the defense lenses to practitioners.

1. Overview of the Defense Lenses

This booklet presents the Software Estimates` Defense Lenses (SWEDeL). It is about a set of lenses that aid software professionals and teams in defending their estimates in the face of pressure for changing them. While it is natural and desirable to change an estimate during estimation sessions because of a better understanding of the problem, the requirements, or about other possible solutions, it is harmful to change them due to pressure alone. Such pressure can be a problem when converting software estimates into commitments with others, like higher managers and customers. Therefore, we resorted to negotiation methods to change the focus from negotiating estimates to negotiating mutually satisfactory commitments. Also, the lenses are useful when others try to impose external and unrealistic commitments to software teams, by helping software professionals to see beyond such imposed commitments and trying to identify paths that satisfy the most the interests of everyone involved.

We designed the lenses based on three interrelated negotiation methods: (a) principled negotiation [1], (b) breakthrough strategy [2], and (c) positive no method [3]. These methods share the objective of promoting negotiations that preserve relationships instead of winning at all costs. The defense lenses are in the format of cards, and Figure 8.1 presents their schematics.

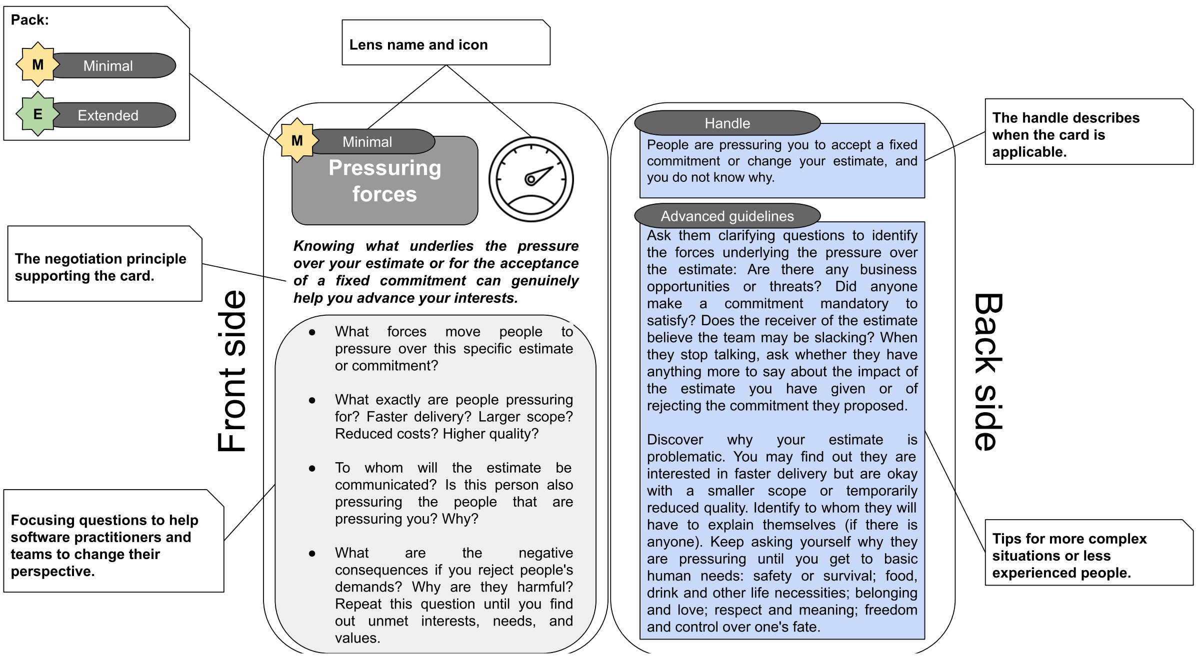


Figure 8.1 - Lens' schematics

On their front side, each card has the lens' name and an icon to represent it. They also are classified according to their recommended pack (as we explain in Section 0). Next, we describe the negotiation principle that supports the lens, which helps software professionals and teams to grasp what it is about. At the gray rounded rectangle, we present a set of focusing questions. They aim at conducting the card user to change their perspective in their path to implement the card's negotiation principle during a real-life communication of their estimates. On the backside, the card comprises a handle, which describes the situations to apply that lens. It helps identify the specific lens we need when using them in isolation instead of using the recommended packs. The card also has advanced guidelines to support less experienced practitioners or help in more complex situations. The next sections present the lenses, organized by their packs. We start with the Minimal Defense Pack, and them move to the Extended Defense Pack.

2. Recommended Card Packs

In this section we present all the advanced defense lenses, classified them according to their recommended card packs. However, each lens can also be used in isolation, by observing their handle. In any case, nothing prevents estimators to combine the lenses from the different packs as they wish, for the specific situations they have in hand. We also present supplementary material for a few of the lenses, in the form of examples.

2.1. Minimal Defense Pack

When to use it:

- You are preparing yourself for providing your estimates and has reasons to believe you will face pressure from people receiving them.
- You provided your estimates and is facing pressure to change them.
- A client or higher manager wants you to commit to a fixed deadline, with a closed scope and with too restricted resources.

The Minimal Defense Pack aims at providing estimators with a set of tools for dealing with a specific pressure episode. The pressure episode can happen during a group estimation session, or when an estimator is providing an individual estimate directly to a client or a higher manager. The pack can help the estimator to deflect from the pressure and to keep their estimates if there is no legitimate reason to change it. Therefore, it is mainly a tactical pack: it helps the estimator to handle concrete episodes in the short run. We recommend estimators to use the pack also **before estimation sessions**, to better **prepare themselves** in case any pressure episode happens. This pack is composed of five lenses: Laddering whys, Choose your battles, Pressuring forces, Assert your estimate, and Candidate commitments.

In Figure 8.2 we present the **Assert your estimate** lens, which aids in communicating your estimate while ensuring it is in the best interest of everyone. Therefore, it works by expressing the estimate to avoid pressure. In an ideal world, this is all you need. In more realistic settings, you are likely to face pressure over your estimates, and you will need the other cards from the minimal defense pack.

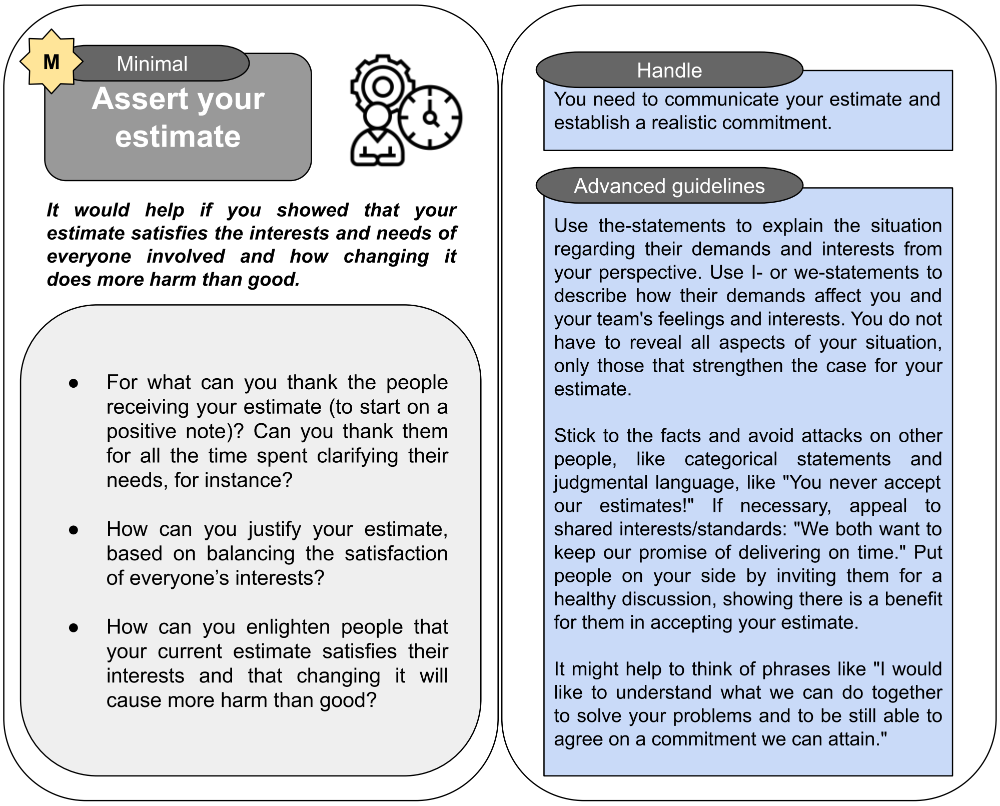


Figure 8.2 - Assert your estimate lens.

The advanced guidelines propose using the-statements, I-statements, and we-statements [3]. They will help you focus on the situation and yourself instead of focusing on the people pressuring you. Table 8.2 presents the definition and examples of these different types of statements.

Table 8.2 - Statements descriptions and examples.

Statement-type	Description	Examples
The-statement	Describe the situation based on the facts plainly and neutrally, instead of blaming the other side for any problems that exist in	• The deadline we agreed on for the last iteration required the delivery of a less polished

Statement-type	Description	Examples
	such a situation (even if they are guilty) [3].	feature, and our current estimate makes space for improving it in this iteration. ● The implementation of the required feature involves data migration to a new format, which takes a lot of effort.
I-statement	Describe your feelings, experience, and interests [3].	● From my experience, I can tell our team won't make more than the four first items in our backlog in this Sprint without compromising their weekend. ● I feel frustrated when there is no room for refactoring in our backlog. ● I believe our team is overwhelmed by the amount of overtime work from the last couple of iterations because the Sprint Backlog included more items than they were capable of handling. Therefore, I think we should leave some room for all of the unexpected events we are facing lately.
We-statement	Describe the joint interests of the parties involved, if uncomfortable with describing your own alone (I-sentences). You can also describe fair standards [3].	● We all want to ensure that the product is delivered on the agreed deadline, right? ● I am sure we both want our employees working according to the law.

When facing a pressure event over a specific estimate, or during an estimation session, you can use the **Pressuring forces** lens, which Figure 8.3 shows. It aims at making you get perspective on what underlies the existing pressure. You may find out that your estimate represents an obstacle for satisfying a relevant need or interest. Therefore, understanding why people pressure for estimate changes is of paramount importance for establishing a mutually satisfactory commitment.

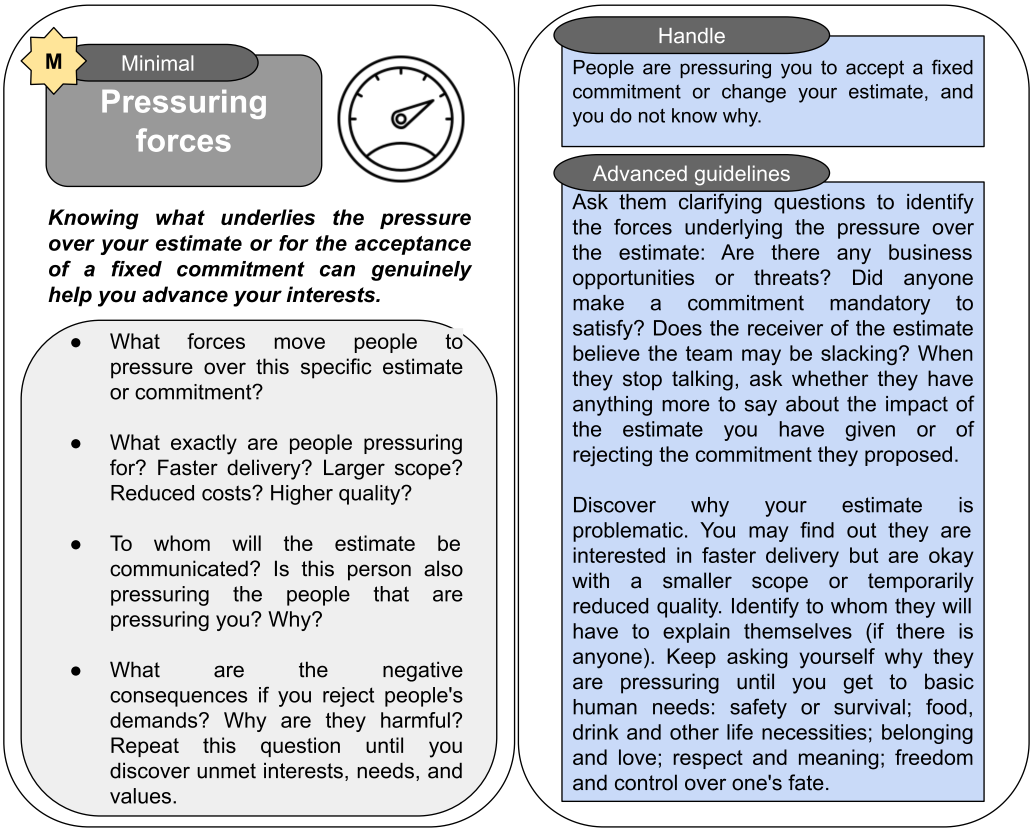


Figure 8.3 - Pressuring forces lens.

Table 8.3 presents a set of detailed clarifying questions that might be useful when using the Pressuring forces lens.

Table 8.3 - Examples of clarifying questions

Clarifying question	Justification	Follow-up
"Does this estimate make project approval unlikely, even though this is a strategic project?"	People may pressure for estimate reductions to make projects more attractive, ensuring they will be approved.	Why is this project strategic? What makes the estimate problematic?
"Did anyone (sales department, an executive) make an external commitment with a customer or with higher management which they need help to satisfy?" Variation: "Is there a preliminary estimate to which someone committed to?"	If there is a previous commitment that the estimate is incompatible with, people will pressure to make sure the commitment is kept.	What exactly is the commitment about? Does it involve a deadline? A fixed scope? A fixed cost? How does the estimate hurt the commitment?
"Is there a business opportunity with specific restrictions we are not aware of, such as a bidding situation?"	If the company is facing a business opportunity, it might be willing to sacrifice the realism of estimates to get it.	What are the restrictions for us to take advantage of such an opportunity? How does the estimate prevent us from taking the opportunity?
"Is there a new client or a higher manager to impress if we provide a faster or cheaper delivery?"	People may pressure to estimate increases or decreases to make a good impression of themselves or the company. For instance, if running a large project is impressive, increases in the estimates are expected. However, if cost-	What would cause a good impression?

Clarifying question	Justification	Follow-up
	efficiency is desirable, lower estimates may be rewarded.	
"Is there a business threat, such as a competitor announcing the launch of a more advanced similar product on the market?"	If the company is facing a business threat, it might be willing to sacrifice the realism of estimates to overcome it.	NA
"Are there any new laws or regulations that with a strict deadline we need to adhere to?"	Laws and regulations usually have a deadline to take effect and attaining the deadlines might be mandatory.	NA
"Does it look like our team can be more productive than this estimate suggests?"	People may believe you are overestimating or unnecessarily padding your estimates. Others think that lower estimates produce healthy pressure, making the team more productive. So, if you do not convince them of the realism of your estimate, they will pressure it.	Break the tasks to clarify the work dimension.
"Is there an internal or external deadline or any other restriction?"	This is only a wild card just in case the other options are unsuccessful in helping to discover what is underlying the pressure over the estimates.	What exactly is the restriction about? In which aspect are we not restricted (costs, quality, scope, deadline)?

Another relevant lens for the moment of pressure is the **Laddering whys,** shown in Figure 8.4. Use it to articulate for yourself and others the interests, needs, and values that are the driving forces for keeping your estimate when facing pressure. We also present a set of examples of using the Laddering whys lens in Table 8.4.

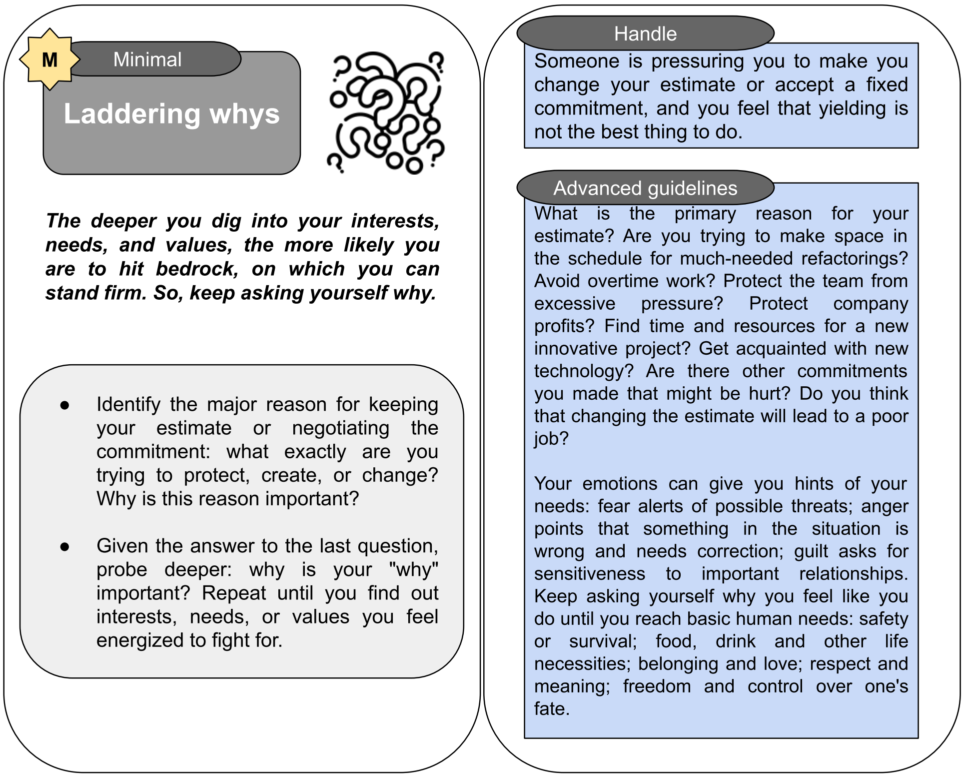


Figure 8.4 - Laddering whys lens.

Table 8.4 - Laddering whys examples.

The reason	Why is this reason important	Why is my "why" important (and corresponding basic needs)
I am trying to make space in the schedule for a much-needed refactoring	Because our product quality is deteriorating beyond acceptable levels. This leads us to a high response time to change requests and more errors in production, impacting our customer satisfaction with our service.	Because our client satisfaction is of utmost importance for keeping our business (safety and survival need)
I am trying to avoid overtime work.	Because I have been working overtime often, neglecting time with my family.	Because my family is the top priority in my life (love and belonging need, freedom, and control over one's fate need).
There are other commitments you made that might be hurt.	Because we need to keep our word to others, no matter what.	Because we must protect our image of a reliable company/reliable people (respect and meaning need).
I think that changing the estimate will lead to a poor job	Because we need to keep a high-quality standard for our product, and we like to have things done right.	Because we must maintain our image of a quality-focused company (respect and meaning need)
I am trying to protect the team from excessive pressure	Because excessive pressure cause quality to drop and people to get unsatisfied, leading to high turnover.	Because we have good people that we care about, and it is not easy to find replacements for them (respect and meaning need, safety and survival need)
I am trying to protect company profit margins	Because we need enough money to keep our payroll and cover our expenses. We also want our shareholders happy and the business attractive for them.	Because we need to provide our people a dignified life (food, drink and other life necessities, safety and survival need)
I am trying to find time and resources for a new innovative project	Because we discovered a much attractive business opportunity we must pursue.	Because we want to expand our business (freedom and control over one's fate need)
I want time to get acquainted with a new technology	Because it can help us to boost our product quality, and I want to keep myself updated	Because I want to be knowledgeable about cutting edge technology and remain attractive for job promotions and opportunities (safety and survival need, freedom and control over one's fate need)

Figure 8.5 shows the **Choose your battles.** It makes you rethink whether you really can keep your estimate in a certain situation or whether it is wiser to change it. In some situations, you need to reassess your decision to say no to pressure, asking yourself whether you have the power, the interest, or the right to do it, as it could lead to an undesirable confrontation [3]. This card brings a bit of strategic thinking to the tense moment of enduring pressure.

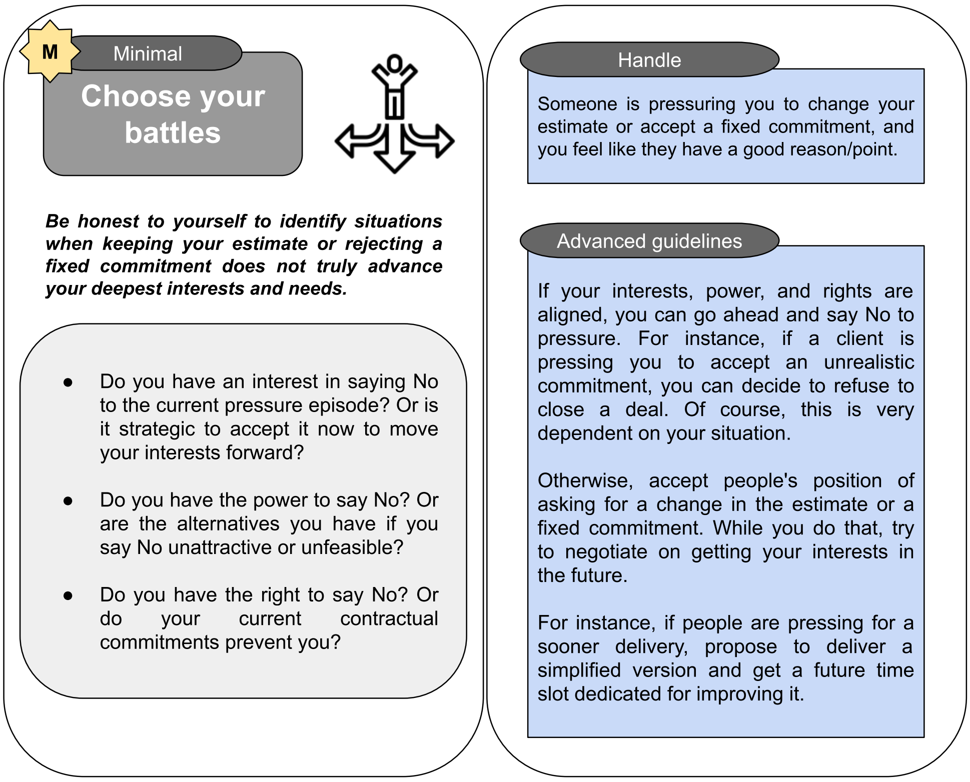


Figure 8.5 - Choose your battles lens.

Table 8.5 presents a set of examples where estimators have no interest, power, or right to say No to pressure and choose to change their estimate or accept an imposed commitment.

Table 8.5 - Examples of situations where estimators have no interest, power, or rights to say No.

You do not have the…	… because…
Interest	All employees, yourself included, will share the profits of this project, which can severely decline because of fines for late deliveries considering the deadline for the entire project.
Power	This is our only client, and we see no new clients in the next two years prospect.
Right	We already signed a contract restricting the budget, schedule, and project scope.

The **Candidate commitment** lens intends to aid in deriving options of mutually satisfactory commitments as we present in Figure 8.6. It is about going beyond your estimate as a position to find alternative paths for keeping the estimate while still trying to accommodate the most interests and needs of everyone involved. In other words, you try to invent options for mutual gains [1].

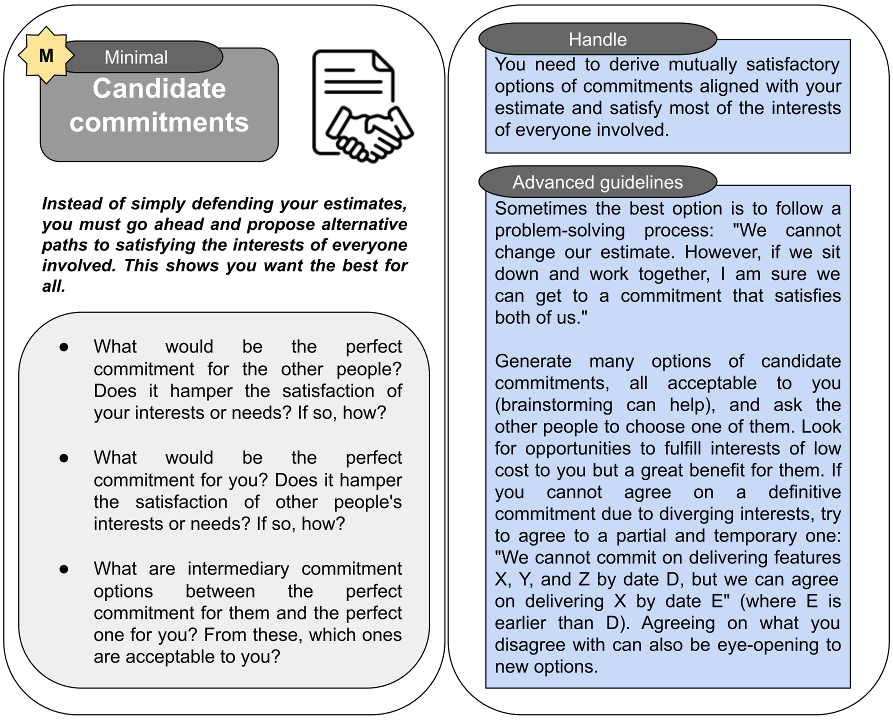


Figure 8.6 - Candidate commitments lens.

In Table 8.6 we present a few options to think of when using the Candidate commitment lens. These options can help you to derive alternative commitments. You can find more detailed examples in McConnell [4].

Table 8.6 – Options for deriving alternative commitments

What are they pressuring for?	Questions to understand further their restrictions	Options to derive alternative commitments
Faster delivery, reduced costs, or higher quality	What are the higher priority features?	- Postpone the delivery of lower priority features [4]
Faster delivery, larger scope, or higher quality	Can we hire additional staff for this project?	- Add more staff to the project, either by hiring new people or by moving people between projects/teams, if it is not too late in the project [4] - Divide the development of the feature with other teams - Keep current staff 100% dedicated to the project [4]
Faster delivery, larger scope, or reduced costs	Can we deliver a simpler version first, and later improve it?	- Simplify the features that will be delivered [4]
Larger scope, reduced costs, or higher quality	Can we be flexible about our schedule?	- Commit to an interval schedule estimate instead of a point estimate [4] - Define revision points in the plan for the schedule project estimate [4]

For a specific pressure episode, the estimator can use the lenses from the Minimal Defense Pack by picking one lens, considering its handle. We suggest the estimator use the Assert your estimate lens every time they are communicating their estimates. If they face pressure to change their estimates, it is time to use the other lenses. The estimator can start with the Pressuring forces lens to understand the other side, and the Laddering whys lens to understand more of their own reasons for keeping the estimate. Next, it is interesting to question themselves regarding their interests/power/rights to keep their estimate with the Choose your battles lens. If the estimator decides they are really keeping their estimates, they can then use the Candidate commitment lens to look for a wise commitment with the other side. If pressure continues even after using the lenses in the Minimal Defense Pack, it is a good time to use the Extended Defense Pack, which we present in the next section.

2.2. Extended Defense Pack

When to use it: You tried to deflect from a pressure episode, but people keep pushing for unjustified changes in your estimates or for the acceptance of an unattainable commitment.

The Extended Defense Pack extends the Minimal Defense Pack and is helpful when you have tried to defend your estimates, but the pressure for changing them continues. It is composed of four additional lenses: Keep strategy, Balcony, Reality test, and Golden bridge.

Figure 8.7 presents the **Keep strategy** lens, which is about stating you have legitimate reasons, sometimes outside your control, for keeping your estimates. That might strengthen your arguments in your favor. Some reasons you can resort to are policies, other commitments, or even your will to do a good job [3]. That might strengthen your arguments in your favor. Besides, you insist on objective criteria [1] to establish a commitment.

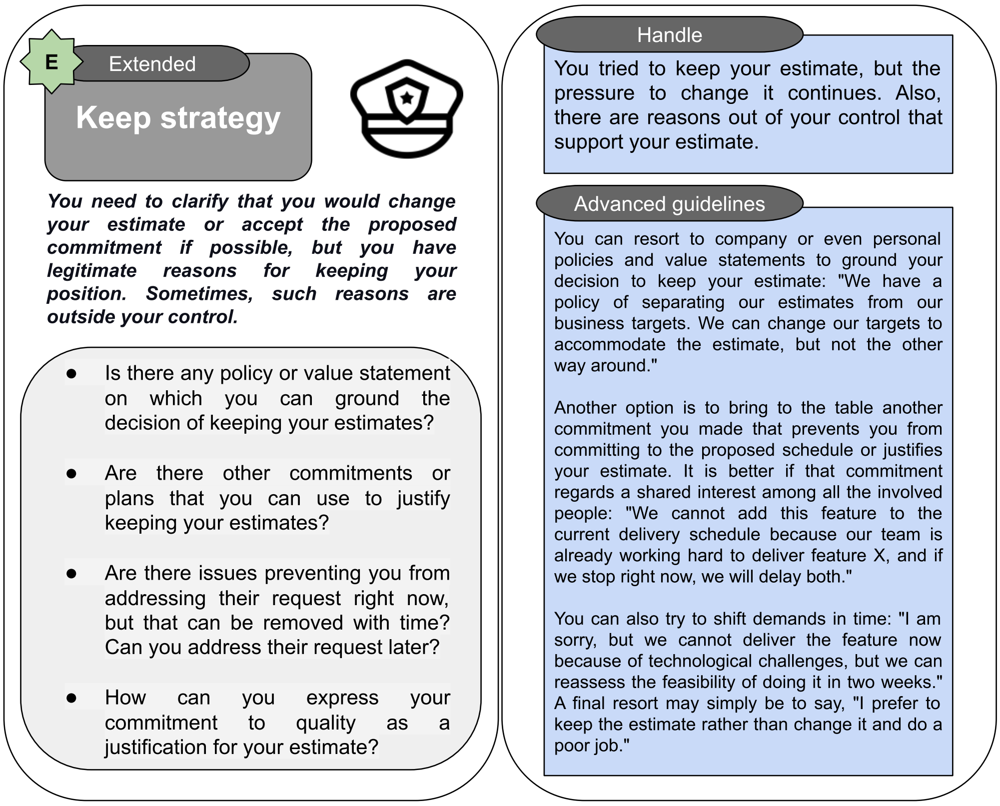


Figure 8.7 - Keep strategy lens.

Figure 8.8 presents the **Perspective taking** lens, useful when pressure is getting stronger. It involves going to a place of perspective, to free yourself from emotions that might impact you negatively, to see more clearly. The other side might be using attacks, stone walls, or tricks to make you change your estimate. An attack tries to intimidate and make you feel uncomfortable; a stone wall is a refusal to budge; a trick will take advantage of your beliefs in their good faith, deceiving you [2].

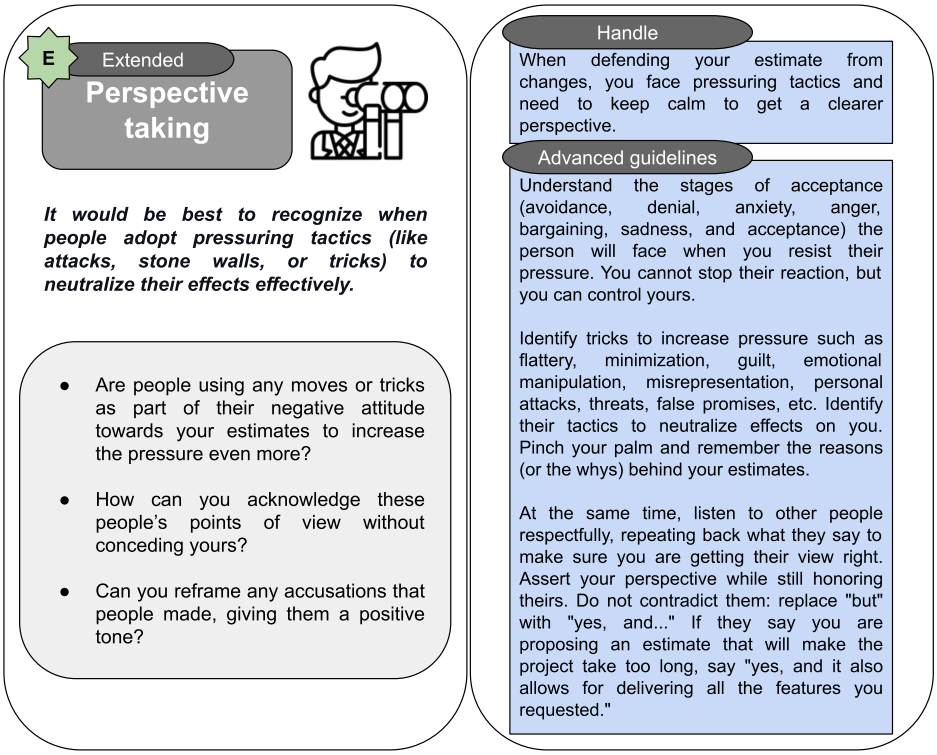


Figure 8.8 – Perspective taking lens.

In Table 8.7 we present examples of different kinds of these tactics in practice. This lens helps you neutralize their effects, by naming them [3], and clarify your thinking so that you do not yield to pressure unnecessarily.

Table 8.7 - Examples of tactics

Tactic	Foundation	Example
Attacks	Based on consequences	“Either you change it or there is no contract!” [2]
	To your proposal	“Your estimates are way out of line!” [2]
	To your credibility	“It looks like you are not so experienced as the rest of your team, uhn…” [2]
	To your authority	“I want to talk with the technical lead, please!” [2]
Stone walls	Previous commitment	“We have already committed with an earlier deadline with the customer. We cannot change that!” [2]
	Final declarations	“It is take it or leave it!” [2]
Tricks and other tactics	Manipulating the data	The other side presents you with a list of features, planning to increase it later on the project. [2]
	Last minute add-on	A last minute new feature is added to the project, right when you thought you had already agreed on the commitment based on the estimates. [2]
	Flattery	“You are the best software team I know! I am sure you can make it to this deadline!” [3]
	Minimization	“But all we need is a small fix on this feature!” [3]

The **Reality test** lens, in Figure 8.9, is an aid for guiding them to change their perspective to the natural and logical consequences of changing the estimate and committing to an unrealistic one. By asking the other side reality-testing questions [2], you can end up showing the point of your estimate. It is based on the idea that asking is better than telling [3]. We provide some examples of reality-testing questions in Table 8.8.

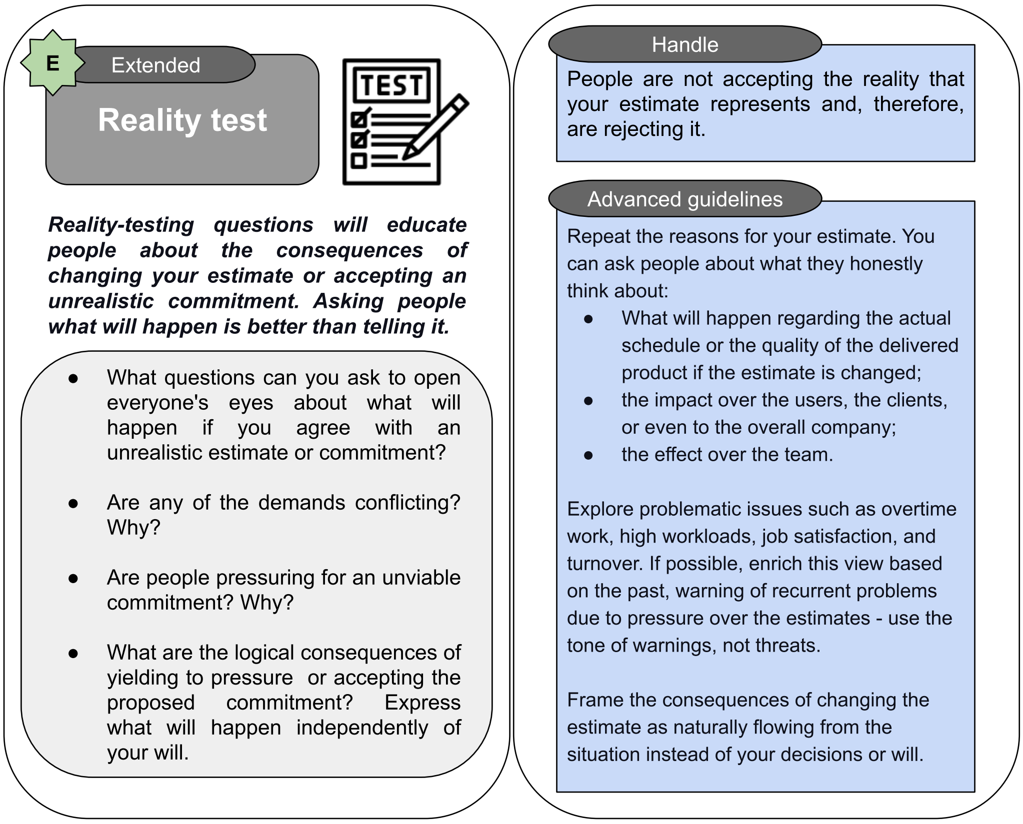


Figure 8.9 - Reality test lens.

Table 8.8 - Examples of reality-testing questions

Impact focus	Example of questions
Schedule-questions	"Ok. Let's say we commit to the deadline you propose, without any changes to our team, to the list of features we have to deliver, and let's suppose we are keeping our high-quality standards. What do you think will happen if someone in our team gets ill?"
Quality-questions	"All right! Let's say we commit to the deadline you propose, without any changes to our team or to the list of features we have to deliver. We won't have time to work on that user interface improvements we have discussed before. Do you think the users will still be willing to use the product without these improvements?"
Users, client, or their company-questions	"Right. Let's say we commit to delivering the product according to your demands. What would happen to your company image if the product failed during your operations because we did not get the time and resources needed to test enough?"
Team-questions	"Fine! Considering we commit to delivering all these features for the next release, how many overtime workhours do you think the team will have to do for the next couple of weeks? How do you think that is going to impact that high turnover issue we have been discussing?"

In any case, if reality-testing questions are not enough, you can also warn the people pressuring you about what can happen, especially about more technical issues that they might be unaware of. The last three focusing questions of this lens aim at helping you to identify issues to warn about and how to do it. A warning is a prediction about inherent consequences that flow from the situation itself and is different from threatening—which is about imposing consequences yourself [3]. The tone is also different: warnings are respectful and show the willingness to collaborate.

Finally, the **Golden bridge** lens helps you build one way so they can retreat from their previous pressure position gracefully, as we show in Figure 8.10. It recognizes they will not accept your estimates if doing it makes them look bad to others. So, you help them to build a bridge from their previous position of pressing for a specific commitment to a new position of accepting a mutually satisfactory one [2].

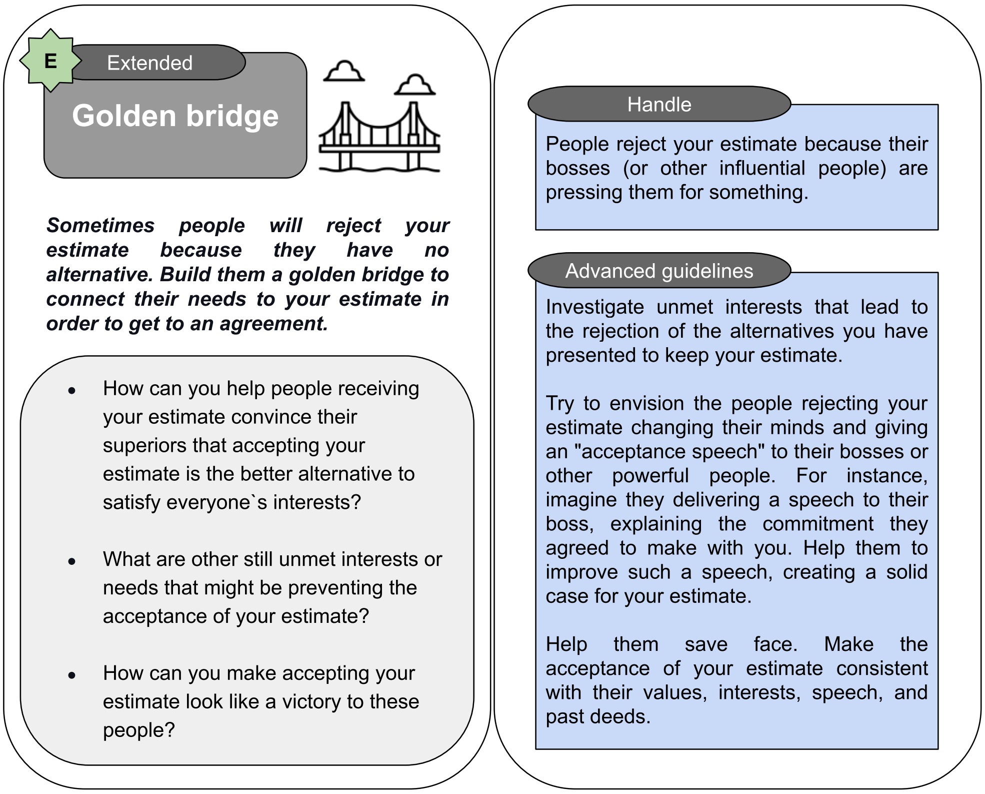


Figure 8.10 - Golden bridge lens.

In summary, if the estimator is suffering from pressuring tactics that could lead them to yield to pressure and change their estimate for no legitimate reason, the first thing to do is to use the Perspective taking lens. Next, the use of the lens depends a lot on the situation at hand, and the estimator needs to consult the handles of the cards to identify the most appropriate one. It may be the case that only one of the remaining lenses applies or all of them.

References

[1] R. Fisher, W. Ury, and B. Patton, Getting to Yes: Negotiating Agreement Without Giving in, 3rd ed. Penguin Books.
[2] W. Ury, Getting Past No: Negotiating in Difficult Situations, Revised edition. Bantam, 2007.
[3] W. Ury, The Power of A Positive No. Hodder & Stoughton, 2012.
[4] S. McConnell, "Politics, Negotiation, and Problem Solving," in Software Estimation: Demystifying the Black Art, Redmond: Microsoft Press, 2006, pp. 259–270.

APPENDIX C– THE BOOKLET (IN PORTUGUESE)

This appendix presents the booklet describing the defense lenses to practitioners in Brazilian portuguese.

1. Visão Geral das Lentes

Esse livreto apresenta as Lentes de Defesa de Estimativas de Software (Software Estimates` Defense Lenses - SWEDeL), um conjunto de lentes para ajudar profissionais e equips de software na defesa de suas estimativas quando são pressionados para mudá-las. Enquanto é natural e desejável mudar uma estimativa durante sessões de estimativa por causa de uma melhor compreensão do problema, dos requisitos ou de outras soluções possíveis, é prejudicial alterá-las por conta de pressão. Essa pressão pode ser um problema quando se converte estimativas de software em compromissos com outras pessoas, como gerentes sêniores e clientes. Portanto, nos voltamos para métodos de negociação a fim de mudar o foco de negociar estimativas para negociar compromissos mutuamente satisfatórios. Além disso, as lentes são úteis quando outras pessoas tentam impor compromissos irrealistas às equipes de software, por ajudar os profissionais a ver além dos compromissos impostos e tentar identificar caminhos que satisfaçam o máximo os interesses de todos.

Nós projetamos as lentes com base em três métodos interrelacionados: (a) *principled negotiation* [1], (b) *breakthrough strategy* [2], e (c) *positive no method* [3]. Esses métodos compartilham o objetivo de promover negociações que preservam os relacionamentos, em vez de se buscar ganhar a qualquer custo. As lentes de defesa estão no formato de cartas, e a Fig. 1 apresenta seu esquema.

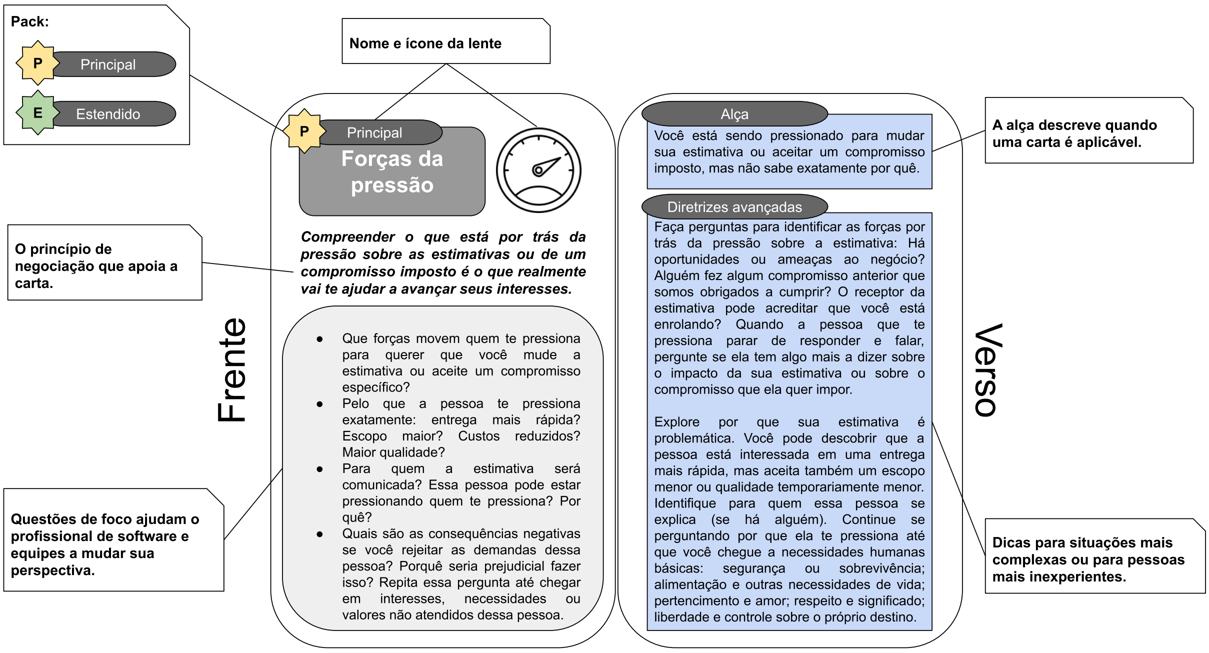


Fig. 1 - Esquema das lentes.

Na frente, cada carta traz o nome e o ícone da lente para representá-la. As lentes estão classificadas de acordo com os seus maços recomendados (como explicamos na Seção 0). Também descrevemos o princípio de negociação que apoia a lente, e que ajuda profissionais e equipes de software a compreender do que a lente trata. No retângulo arredondado cinza, apresentamos um conjunto de questões de foco, que objetivam conduzir o usuário da carta a mudar sua perspectiva em seu caminho na implementação do princípio de negociação da lente durante um episódio real de comunicação de suas estimativas. No verso, a carta é composta de uma alça, que descreve as situações em que se aplica e ajuda a identificar a lente específica isoladamente, quando não usamos os maços recomendados. A carta também tem diretrizes avançadas para apoiar profissionais menos experientes ou para ajudar em situações mais complexas. Nas próximas seções, apresentamos as lentes avançadas, organizadas por maço. Começamos com o Maço Principal de Defesa e seguimos para o Maço Estendido de Defesa.

2. Maços Recomendados

Nessa seção apresentamos todas lentes avançadas de defesa, classificando-as de acordo com seus maços recomendados. Contudo, cada lente também pode ser usada isoladamente, considerando sua alça. Adicionalmente, fornecemos fluxos sugeridos de uso das lentes em cada maço. De qualquer forma, nada impede que os estimadores combinem as lentes como preferirem, inclusive usando cartas de maços diferentes, considerando as situações específicas que têm em mãos. Também fornecemos material complementar para algumas das lentes, na forma de exemplos.

2.1. Maço Principal de Defesa

Quando usar:

- Você está se preparando para fornecer suas estimativas e tem razões para acreditar que será pressionado pelas pessoas que vão recebê-las.
- Você forneceu suas estimativas e está sendo pressionado para mudá-las.
- Um cliente ou gerente sênior quer que você se comprometa com um prazo fixo, com um escopo fechado e com recursos muito restritos.

O Maço Principal de Defesa visa fornecer aos estimadores com um conjunto de lentes para lidar com episódios específicos de pressão. Tais episódios podem acontecer durante uma sessão de estimativas em grupo, ou quando um estimador fornece uma estimativa individual diretamente para um cliente ou gerente sênior. O maço pode ajudar o estimador a desviar da pressão e manter suas estimativas, se não há razões legítimas para alterá-las. Portanto, é um maço tático: ajuda o estimador a lidar com episódios concretos no curto prazo. Recomendamos que os estimadores o usem também antes das sessões de estimativa, para se prepararem melhor no caso de ocorrer pressão. Esse maço é composto de cinco lentes: Afirme sua estimativa, Forças da pressão, Porquês encadeados, Escolha suas batalhas e Compromissos candidatos.

Na Fig. 2 apresentamos a Lente **Afirme sua estimativa**, que apoia a comunicação de suas estimativas garantindo que é pelo bem de todos. Assim, a lente funciona por guiar o estimador a expressar sua estimativa evitando a pressão. Em um mundo ideal, isso seria tudo que se precisa. No entanto, em ambientes mais realistas, você provavelmente vai enfrentar pressões sobre sua estimativa, e precisará das outras cartas desse maço.

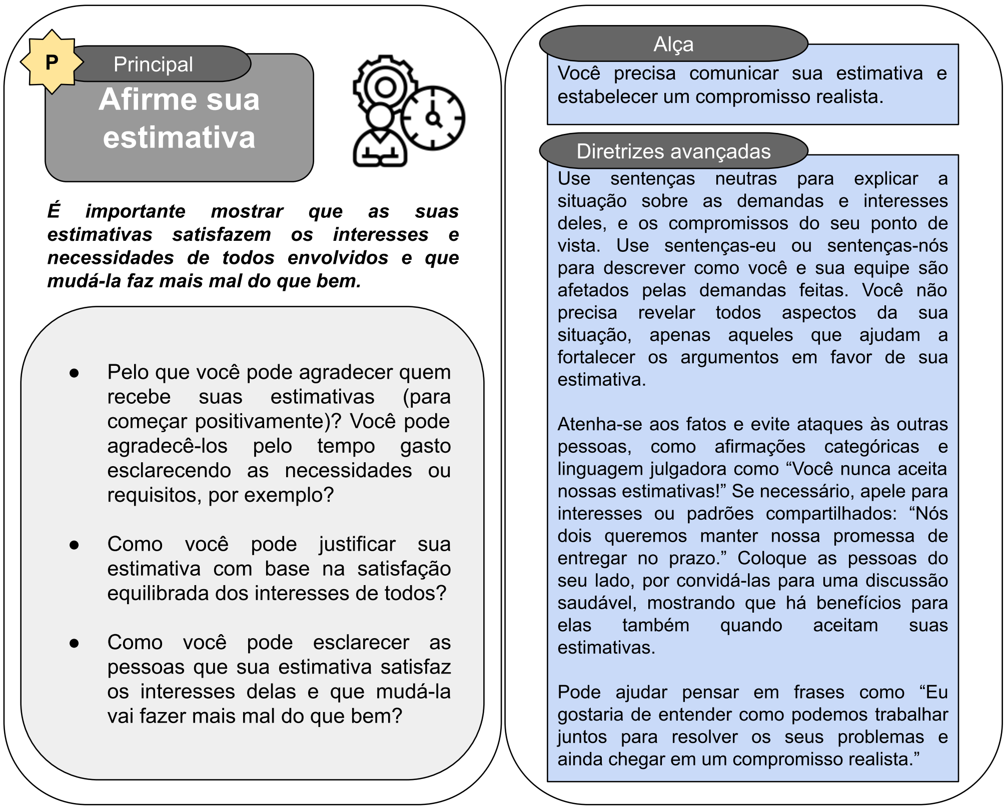


Fig. 2 – Lente Afirme sua estimativa.

As diretrizes avançadas propõem o uso de sentenças neutras, sentenças-eu e sentenças-nós [20]. Essas sentenças te ajudarão a manter o foco na situação e em si mesmo(a), em vez de focar nas pessoas que te pressionam. Na Tabela 1 são mostrados definições e exemplos desses diferentes tipos de sentenças.

Tabela 1 - Descrição e exemplos de sentenças.

Tipo de sentença	Descrição	Exemplos
Sentença neutra	Descreva a situação com base nos fatos de forma clara e neutral, em vez de culpar outros por qualquer problema que exista (mesmo que sejam culpados) [20].	• O prazo que combinamos para a iteração passada exigiu a entrega de uma funcionalidade de forma menos polida, e nossa estimativa atual tenta arranjar tempo para melhorá-la na próxima iteração. • A implementação da funciondalidade solicitada envolve a migração dos dados para um novo formato, e isso vai requerer um esforço enorme.
Sentença-eu	Fale do que você sente, da sua experiência, e dos seus interesses [20].	• Na minha experiência, posso dizer que nosso time não vai entregar mais do que os quatro primeiros itens do backlog nessa Sprint a não ser que entre fins de semana adentro. • Me sinto frustrado quando a gente não tem tempo no cronograma para fazer um pouco de refatoração. • Acredito que nosso time está estafado com tantas horas extras que tiveram que fazer nas últimas iterações, pois o Sprint Backlog tinha mais itens do que eram capazes de entregar. Por

		isso, penso que devíamos deixar um tempo para os eventos inesperados que estamos enfrentando ultimamente.
Sentença-nós	Fale dos interesses conjuntos que você e as outras pessoas (inclusive quem te pressiona) têm, caso se sinta desconfortável de falar só dos seus interesses (com as sentenças-eu). Outra alternativa é descrever critérios justos [20].	• Todos queremos garantir que o produto vai ser entregue dentro do prazo que acordarmos, certo? • Tenho certeza que todos nós queremos nossos colaboradores trabalhando de acordo com a lei.

Quando enfrentar pressão sobre uma estimativa específica, ou durante uma sessão de estimativa, você pode usar a Lente **Forças da pressão**, mostrada na Fig. 3. Essa lente objetiva te fazer obter perspectiva sobre o que está por trás da pressão existente. Você pode descobrir que sua estimativa representa um obstáculo para uma necessidade ou interesse relevante. Assim, ao compreender por que as pessoas estão pressionando por uma mudança na estimativa é essencial para estabelecer um compromisso mutuamente satisfatório.

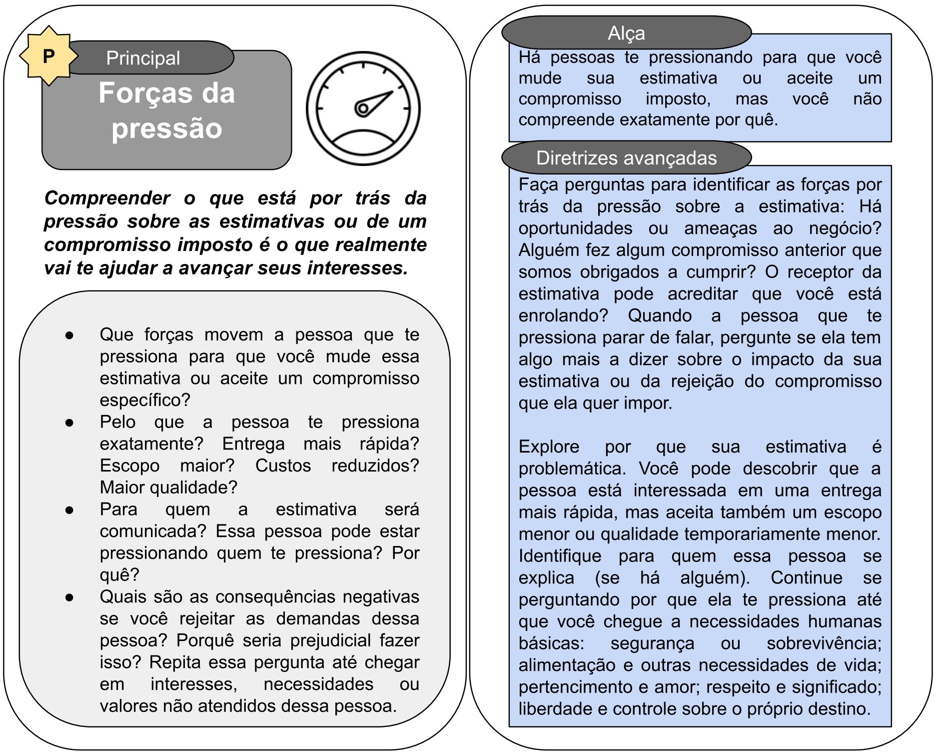


Fig. 3 - Lente Forças da pressão.

A Tabela 2 apresenta uma série de questões de esclarecimento [20] que você pode fazer ao lado que pressiona, quando usa a Lente Forças da pressão.

Tabela 2 - Exemplos de questões esclarecedoras.

Questões de esclarecimento	Justificativa	*Follow-up*
"Essa estimativa dificulta a aprovação do projeto, apesar de se tratar de um projeto estratégico?"	As pessoas podem pressionar por reduções nas estimativas para fazer os projetos parecerem mais atrativos, e garantir que serão aprovados.	Por que o projeto é estratégico? O que faz a estimativa problemática?
"Alguém (departamento de vendas, um executivo) fez um compromisso externo com um cliente ou com a gerência sênior que precisa ser cumprido?" Variação: "Há uma estimativa preliminar com que alguém se comprometeu?"	Se há um compromisso anterior com o qual a estimativa é incompatível, as pessoas irão pressionar para garantir que o compromisso é cumprido.	Sobre o que exatamente é o compromisso? Envolve um prazo? Um escopo fixo? Um custo fixo? Como a estimativa fere o compromisso?
"Há alguma oportunidade de negócio com restrições específicas que desconhecemos, como uma oportunidade de licitação?"	Se a organização tem uma oportunidade de negócio, as pessoas podem estar dispostas a sacrificar o realismo das estimativas para aproveitá-la.	Quais são as restrições para que aproveitemos a oportunidade? Como a estimativa impede que consigamos isso?
"Há um cliente novo ou um gerente sênior a impressionar se entregarmos mais rapidamente ou fizermos algo mais econômico?"	As pessoas podem pressionar por aumentos ou reduções de estimativa para causar uma boa impressão. Por exemplo, se é bom gerenciar um projeto grande, pode-se esperar aumentos nas estimativas. No entanto, se eficiência de custo é desejável, estimativas menores podem ser recompensadas.	O que causaria uma boa impressão?
"Há uma ameaça de negócio, como um competidor anunciando o lançamento de um produto similar mais avançado?"	Se a organização está diante de uma ameaça de negócio, pode sacrificar o realismo das estimativas para superá-la.	NA
"Há alguma lei ou regulamentação a ser lançada com um prazo estrito e que precisamos obedecer?"	Leis e regulamentações geralmente têm um prazo para entrar em vigor e cumprir os prazos pode ser obrigatório.	NA
"Parece que nossa equipe pode ser mais produtiva do que nossa estimativa sugere?"	As pessoas podem achar que você está superestimando ou colocando gordura desnecessária em suas estimativas. Outras pessoas acham que estimativas menores produzem uma pressão saudável, fazendo a equipe mais produtiva. Então, se você não as convencê-las do realismo da sua estimativa, elas irão pressioná-lo.	Quebre as tarefas para esclarecer a dimensão do trabalho.
"Há algum prazo interno ou externo, ou qualquer outra restrição?"	Esse é apenas uma carta na manga para o caso de as outras questões esclarecedoras não terem ajudado muito a descobrir o que está por trás da pressão sobre as estimativas.	Sobre o que exatamente é a restrição? Em qual aspecto não há restrições (custo, qualidade, escopo, prazo)?

Outra lente relevante para o momento da pressão é a de **Porquês encadeados**, mostrada na Fig. 4. Use-a para articular para si mesmo e para os outros os interesses, necessidades e valores que são as forças motrizes para manter suas estimativas quando enfrenta pressão. Para isso, se pergunte o porquê por trás das razões de manter sua estimativa até chegar nessas forças

[20]. A Tabela 3 apresenta também um conjunto de exemplos para ajudar a usar a Lente Porquês encadeados.

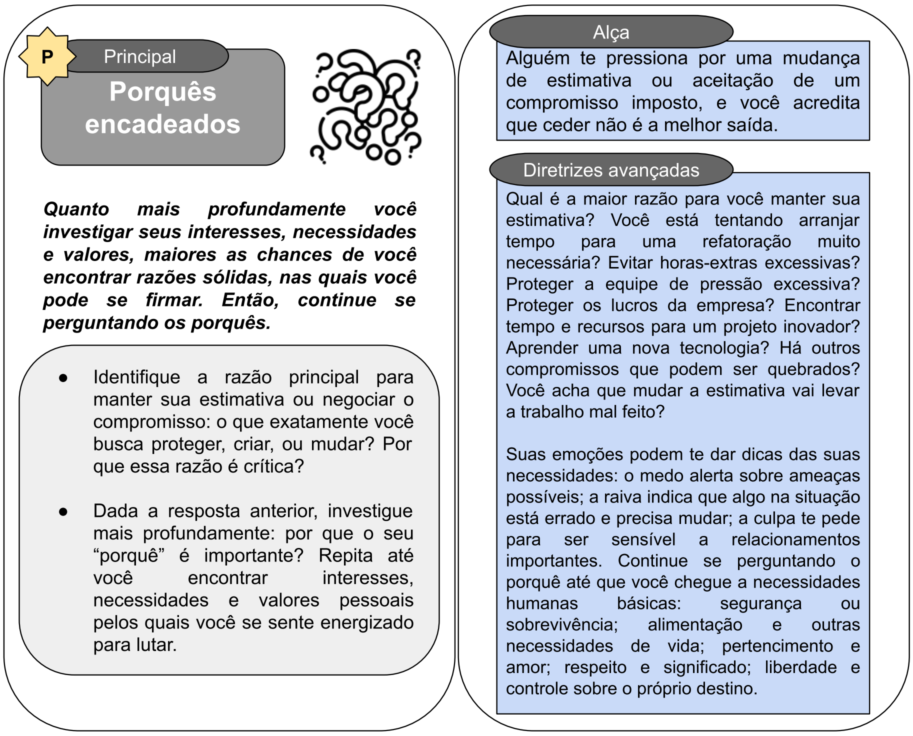


Fig. 4 - Lente Porquês encadeados.

Tabela 3 - Exemplos de razões da Lente Porquês encadeados.

A razão	Por que essa razão é crítica	Por que meu "porquê" é importante (e necessidades básicas correspondentes)
Estou tentando criar espaço no cronograma para uma refatoração muito necessária	Porque a qualidade do produto está deteriorando além de níveis aceitáveis. Isso nos leva a um maior tempo de resposta a pedidos de mudança e a mais erros em produção, impactando a satisfação do nosso cliente com nossos serviços.	Porque a satisfação do nosso cliente é de importância máxima para manter nosso negócio (necessidade de segurança e sobrevivência)
Estou tentando evitar horas-extras em excesso.	Porque tenho trabalho muitas horas-extras ultimamente, negligenciando minha família.	Porque minha família é a maior prioridade na minha vida (necessidade de amor e pertencimento, necessidade de liberdade e controle do próprio destino).
Há outros compromissos feitos que podem ser prejudicados.	Porque precisamos manter nossa palavra para os outros, não importam as circunstâncias.	Porque precisamos proteger nossa imagem de uma organização/pessoa confiável (necessidade de respeito e significado).
Acho que mudar a estimativa nos levará a um trabalho mal-feito.	Porque precisamos manter um padrão alto de qualidade para nosso produto, e gostamos que as coisas sejam bem feitas.	Porque precisamos manter nossa imagem de uma empresa focada em qualidade (necessidade de respeito e significado).
Estou tentando proteger a equipe de pressão excessiva.	Porque a pressão excessiva pode levar a piora na qualidade e as pessoas a ficarem insatisfeitas, aumentando a rotatividade.	Porque temos pessoas bem qualificadas, com as quais nos preocupamos, e é difícil encontrar bons profissionais para substituí-las (necessidade de respeito e significado; necessidade de segurança e sobrevivência)

A razão	Por que essa razão é crítica	Por que meu “porquê” é importante (e necessidades básicas correspondentes)
Estou tentando proteger as margens de lucro da empresa.	Porque precisamos de dinheiro suficiente para manter a folha de pagamentos e cobrir nossas despesas. Também queremos manter nossos acionistas felizes e o negócio atrativo para eles.	Porque precisamos dar às pessoas uma vida digna (necessidade de alimentação e outros itens básicos; necessidade de segurança e sobrevivência).
Estou tentando encontrar tempo e recursos para um projeto inovador.	Porque descobrimos uma oportunidade de negócio muito atrativa que devemos perseguir.	Porque queremos expandir nossos negócios (necessidade de liberdade e controle do próprio destino).
Eu quero ter tempo de aprender uma nova tecnologia.	Porque essa tecnologia pode nos ajudar a impulsionar a qualidade do produto, e eu quero me manter atualizado.	Porque eu quero estar bem informado das tecnologias de ponta e me manter atrativo para promoções e oportunidades de trabalho (necessidade de segurança e sobrevivência; necessidade de liberdade e controle do próprio destino).

A Lente **Escolha suas batalhas** te faz repensar se você realmente pode manter sua estimativa em uma dada situação ou se é mais sábio mudá-la, te ajudando a focar nos seus interesses reais [18]. Em algumas situações, você precisa reavaliar sua decisão de dizer não à pressão, se perguntando se realmente tem o poder, o interesse ou o direito de dizer para fazê-lo, uma vez que pode levar a um confronto indesejável [20]. Essa carta traz um pouco de pensamento estratégico para o momento tenso de enfrentar a pressão. A Fig. 5 mostra a lente.

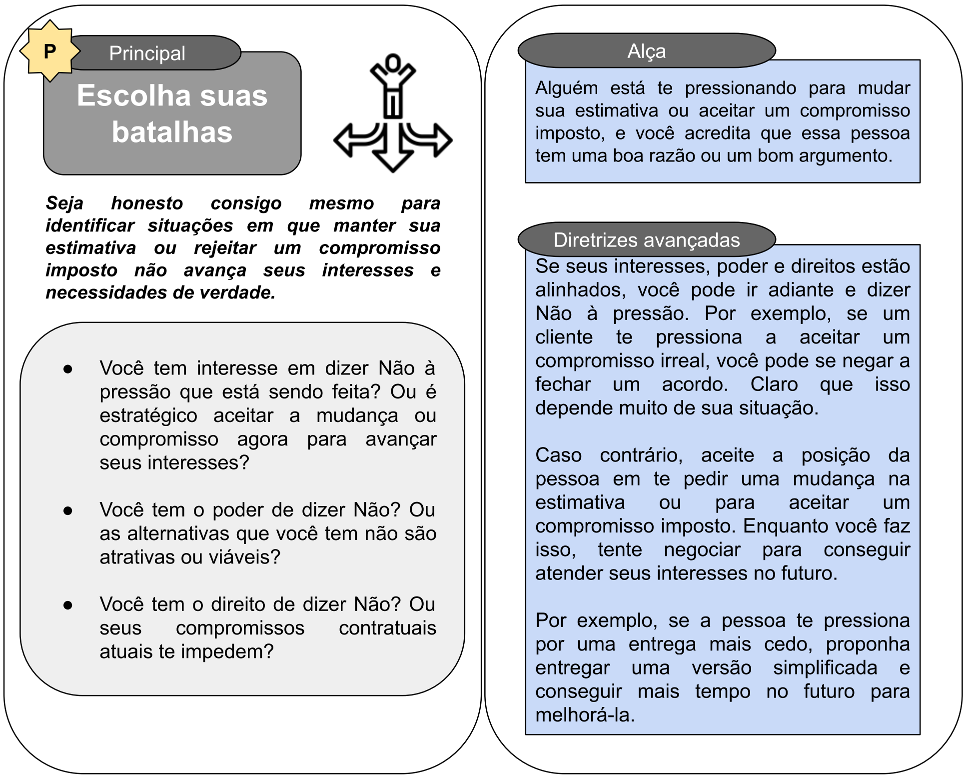


Fig. 5 - Lente Escolha suas batalhas.

A Tabela 3 apresenta um conjunto de exemplos onde os estimadores não têm o interesse, o poder, ou o direito de dizer Não à pressão e escolhem mudar sua estimativa ou aceitar um compromisso imposto.

Tabela 3 – Examplos de situações onde os estimadores não têm interesse, poder ou direito de dizer Não.

Você não tem o...	… porque…
Interesse	Todos colaboradores, inclusive você, vão dividir os lucros desse projeto, que podem cair severamente por conta de multas devido a entregas atrasadas, considerando o prazo para o projeto.
Poder	Esse é seu único cliente, e você não enxerga a possibilidade de ter novos clientes pelos próximos dois anos.
Direito	Já assinamos um contrato restringindo o orçamento, cronograma e escopo do projeto.

A Lente **Compromissos candidatos** tem a intenção de ajudar os estimadores a derivar opções de compromissos mutuamente satisfatórios, como mostra a Fig. 6. O uso da lente envolve ir além de sua estimativa como uma posição para encontrar caminhos alternativos para mantê-la enquanto se tenta acomodar os interesses e necessidades dos envolvidos o máximo possível. Em outras palavras, você busca inventar opções para promover ganhos mútuos [18].

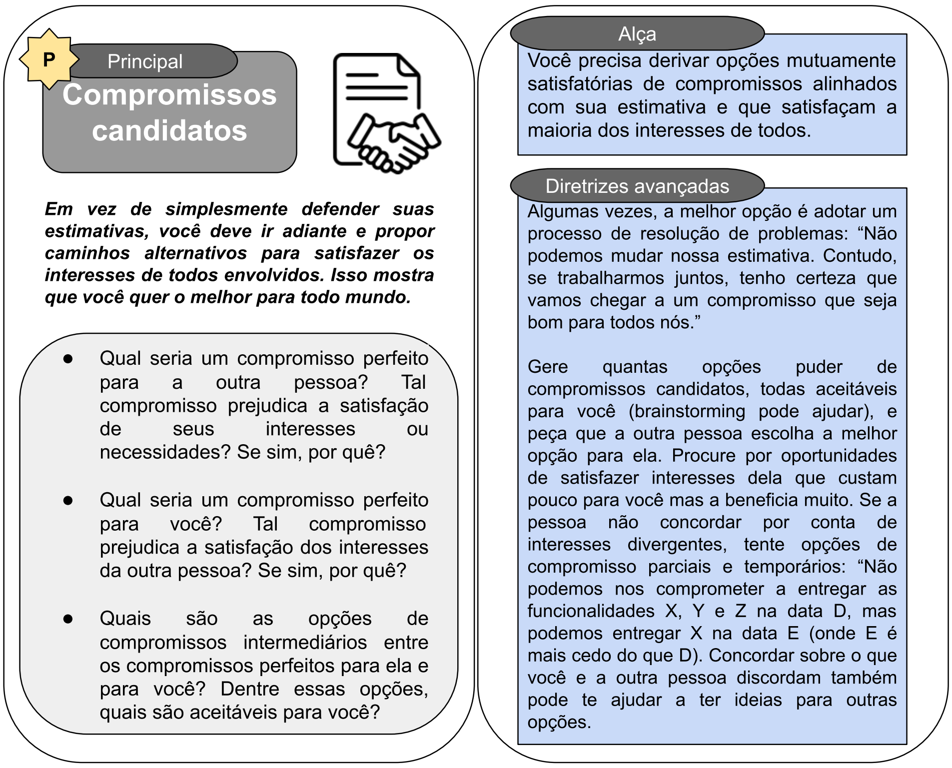


Fig. 6 - Lente Compromissos candidatos.

Na Tabela 4 apresentamos algumas opções que você pode considerar quando usar essa lente, a fim de tentar derivar outras alternativas de compromissos. Você pode encontrar exemplos mais detalhados em McConnell [17].

Tabela 4 - Opções para derivar compromissos candidatos.

Pelo que você é pressionado?	Questões para compreender as restrições	Opções para derivar compromissos alternativos
Entrega mais rápida, custos reduzidos, ou maior qualidade	Quais são as funcionalidades de maior prioridade?	- Adie a entrega das funcionalidades de menor prioridade [17]
Entrega mais rápida, escopo maior, ou maior qualidade	Podemos contratar pessoas a mais para esse projeto?	- Adicione mais pessoas ao projeto, seja com novas contratações ou por trocas entre projetos/equipes, se não for tarde demais no projeto [17] - Divida o desenvolvimento da funcionalidade com outras equipes [17] - Mantenha a equipe atual 100% dedicada ao projeto [17]
Entrega mais rápida, escopo maior, ou custos reduzidos	Podemos entregar uma versão mais simples primeiro, e depois melhorá-la?	- Simplifique as funcionalidades que vão ser entregues [17]
Escopo maior, custos reduzidos, ou maior qualidade	Podemos ser flexíveis quanto ao cronograma/prazo?	- Se comprometa com uma estimativa de cronograma intervalar em vez de uma estimativa pontual [17] - Defina pontos de revisão no plano para a estimativa de prazo [17]

Para um episódio específico de pressão, o estimador pode escolher uma lente desse maço a partir de sua alça. Sugerimos que o estimador comece usando a Lente Afirme sua estimativa todas as vezes que for comunicá-las. Se enfrentar pressão para mudá-las, é hora de usar as outras lentes. O estimador pode começar usando a Lente Forças da pressão para compreender o lado que pressiona, a de Porquês encadeados, para compreender mais de suas próprias razões para manter sua estimativa. Em seguida, é interessante que o estimador se questione sobre seus interesses/poder/direitos de manter a sua estimativa com a Lente Escolha suas batalhas. Se decidir realmente manter sua estimativa, o estimador pode então usar a Lente Compromissos candidatos para buscar um compromisso sábio com o lado que pressiona. Se tudo isso falhar, pode ser uma boa hora para usar as lentes do Maço Estendido de Defesa, apresentadas na seção a seguir.

2.2. Maço Estendido de Defesa

Quando usar: Você tentou desviar de um episódio de pressão específico, mas as pessoas continuam te pressionando por mudanças não justificadas na sua estimativa ou pelo aceite de um compromisso irrealista.

O Maço Estendido de Defesa é uma extensão do Maço Principal de Defesa e é útil quando você já tentou defender suas estimativas, mas a pressão continua. É composto de quatro lentes avançadas adicionais: Estratégia de guarda, Perspectiva, Teste de realidade e Ponte de ouro.

A Lente **Estratégia de guarda** é mostrada lente na Fig. 7. Envolve informar que você tem razões legítimas, muitas vezes fora do seu controle, para manter sua estimativa. Você pode recorrer a políticas, outros compromissos ou até ao seu desejo de fazer um bom trabalho como uma dessas razões [20]. Isso pode fortalecer os argumentos a seu favor. Além disso, você também estará insistindo em usar critérios objetivos [18] para estabelecer um compromisso.

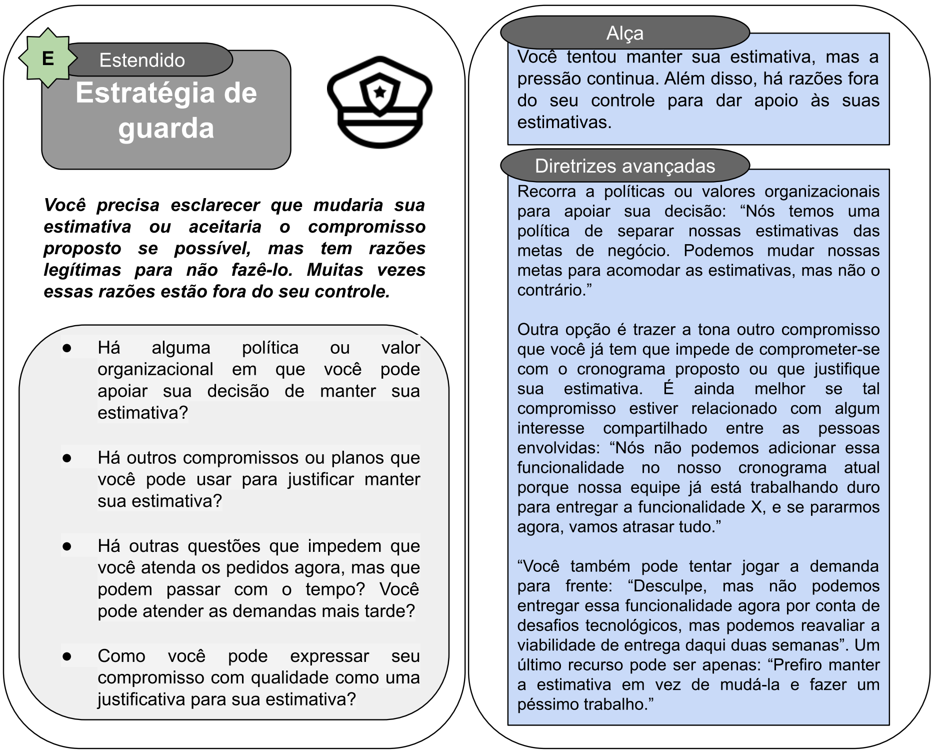


Fig. 7 - Lente Estratégia de guarda.

Na Fig. 8 apresentamos a Lente **Ganho de perspectiva**, útil quando a pressão está se tornando mais forte. Essa lente envolve ir para um lugar de perspectiva, para se libertar das emoções que podem te impactar negativamente, a fim de ver mais claramente. O lado que pressiona pode estar a usar ataques, "muralhas de pedra", ou truques para te fazer mudar sua estimativa. Um ataque tenta te intimidar e te fazer sentir desconfortável; uma “muralha de pedra” é uma recusa em ceder o mínimo que seja; um truque se aproveita de suas crenças na boa vontade do outro lado, te enganando [19].

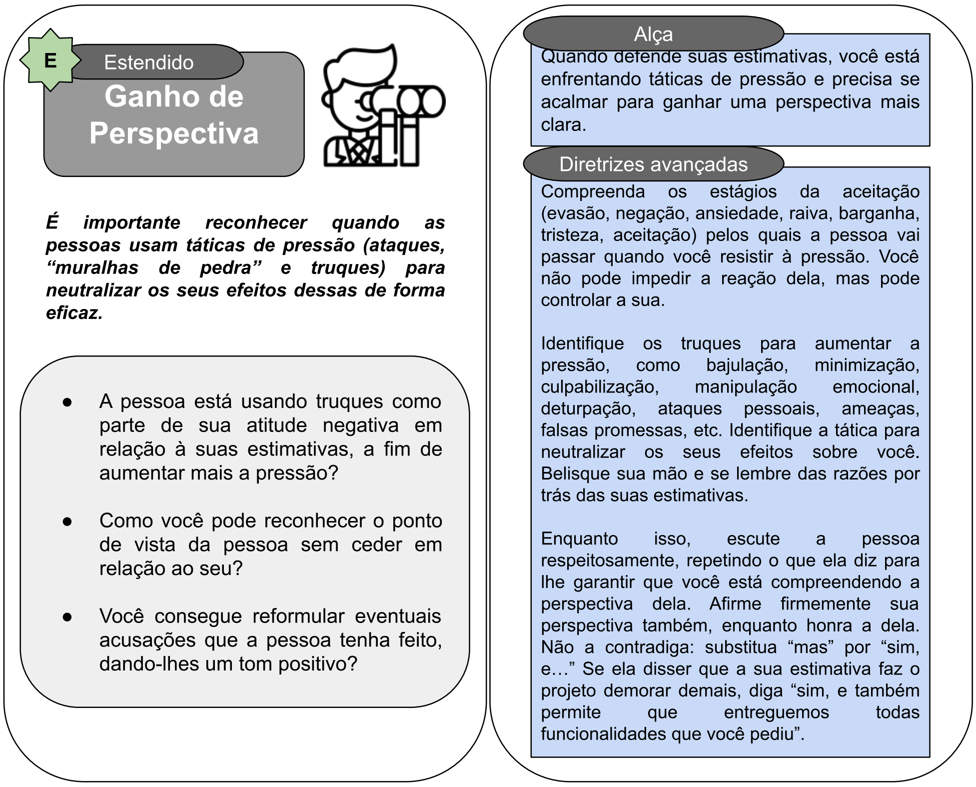


Fig. 8 - Lente Perspectiva.

Na Tabela 5 mostramos exemplos de diferentes tipos dessas táticas na prática. Essa lente te ajuda a neutralizar os efeitos dessas táticas, dando-lhes os nomes corretos [20], e esclarecendo seus pensamentos para que você não ceda à pressão desnecessariamente.

Tabela 5 - Exemplo das táticas.

Tática	Base	Exemplo
Ataques	Baseada nas consequências	“Ou você muda ou não assinamos o contrato!” [19]
	À sua proposta	“Suas estimativas estão bem fora, hein!” [19]
	À sua credibilidade	“Parece que você não tem tanta experiência quanto o resto da equipe, né?” [19]
	À sua autoridade	“Eu quero falar com o líder técnico, por favor!” [19]
“Muralhas de pedra”	Compromisso anterior	“Nós já nos comprometemos com um prazo mais cedo com o cliente. Não dá para mudar isso!” [19]
	Declarações decisivas	“É pegar ou largar!” [19]
Truques e outras táticas	Manipulação dos dados	O outro lado te dá uma lista de funcionalidades, planejando aumentá-la depois no projeto. [19]
	Adição de último minuto	Uma funcionalidade é adicionada de ultimo minute no projeto, logo depois que você achava que já

Tática	Base	Exemplo
		tinham firmado um compromisso com base nas estimativas. [19]
	Bajulação	“Vocês formam a melhor equipe que eu conheço! Tenho certeza que vão conseguir entregar nesse prazo!” [20]
	Minimização	“Mas tudo que precisamos é uma pequena alteração nessa funcionalidade!” [20]

A Lente **Teste de realidade** é uma ajuda para guiar o lado que pressiona na sua perspectiva sobre as consequências naturais e lógicas de mudar a estimativa ou de firmar um compromisso irrealista. Ela é apresentada na Fig. 9. Por perguntar ao lado que pressiona questões de teste de realidade [19], você pode acabar lhes mostrando o ponto da sua estimativa. Isso se baseia na ideia de que perguntar é mais eficaz do que dizer o que vai acontecer [20]. Também apresentamos alguns exemplos de questões de teste de realidade na Tabela 6.

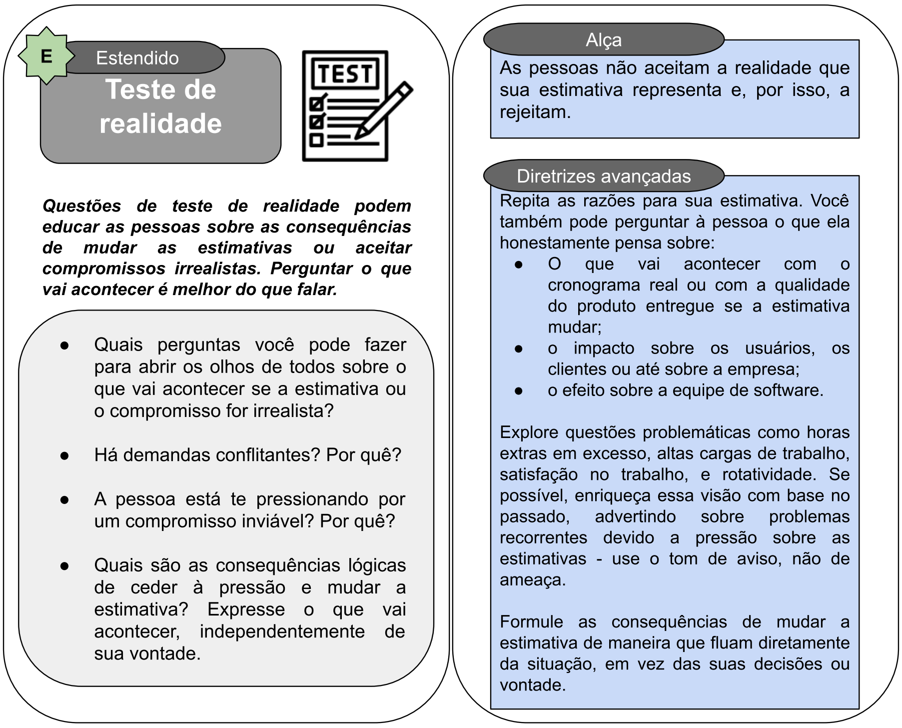


Fig. 9 - Lente Teste de realidade.

Tabela 6 - Exemplos de questões de teste de realidade.

Foco do impacto	Exemplos de questões
Questões sobre o prazo/cronograma	“Ok. Vamos dizer que firmamos um compromisso com o prazo que você propõe, sem nenhuma mudança na equipe nem no escopo a entregar. Vamos supor também que vamos manter nosso padrão de alta qualidade. O que você acredita que vai ocorrer se alguém da equipe ficar doente?”
Questões sobre qualidade	“Certo! Vamos dizer que firmamos um compromisso com o prazo que você propõe, sem nenhuma mudança na equipe nem no escopo

Foco do impacto	Exemplos de questões
	a entregar. Não vamos ter tempo de trabalhar nas melhorias da interface do usuário que já conversamos antes. Você acha que os usuários vão continuar querendo usar o produto sem essas melhorias?”
Questões sobre usuários, clients, ou a organização	“Certo… vamos supor que nos comprometemos a entregar o produto conforme suas demandas. O que vai acontecer com a imagem da organização se o produto falhar durante a operação porque não tivemos tempo e recursos suficientes para testar o que era necessário?”
Questões sobre a equipe	“Tudo bem! Supondo que nos comprometemos em entregar todas essas funcionalidades na próxima iteração, quantas horas-extras você acha que a equipe vai ter que cumprir nas próximas semanas? Como você acha que isso vai impactar aquela questão de rotatividade que estivemos discutindo?”

De qualquer forma, se questões de teste de realidade não forem o suficiente, você também pode avisar as pessoas que pressionam sobre o que pode acontecer, especialmente em relação a questões mais técnicas que eles podem desconhecer. As três últimas questões de foco dessa lente te ajudam a identificar questões para avisar. Um aviso é uma previsão sobre consequências inerentes que fluem da situação em questão e são diferentes de ameaças—que é sobre você mesmo impor consequências à outra pessoa [20]. O tom também é diferente: avisos são respeitosos e mostram o desejo de colaborar.

Finalmente, a Lente **Ponte de ouro** te ajuda a construir um caminho pelo qual o lado que pressiona pode graciosamente recuar da sua posição de pressionar, como mostra a Fig. 10. Essa lente reconhece que o lado que pressiona não vai aceitar suas estimativas se isso os fizer parecer fracos para outros. Então, ajude-os a construir uma ponte de ouro de sua posição anterior para uma posição de aceitar um compromisso que seja mutuamente satisfatório [19].

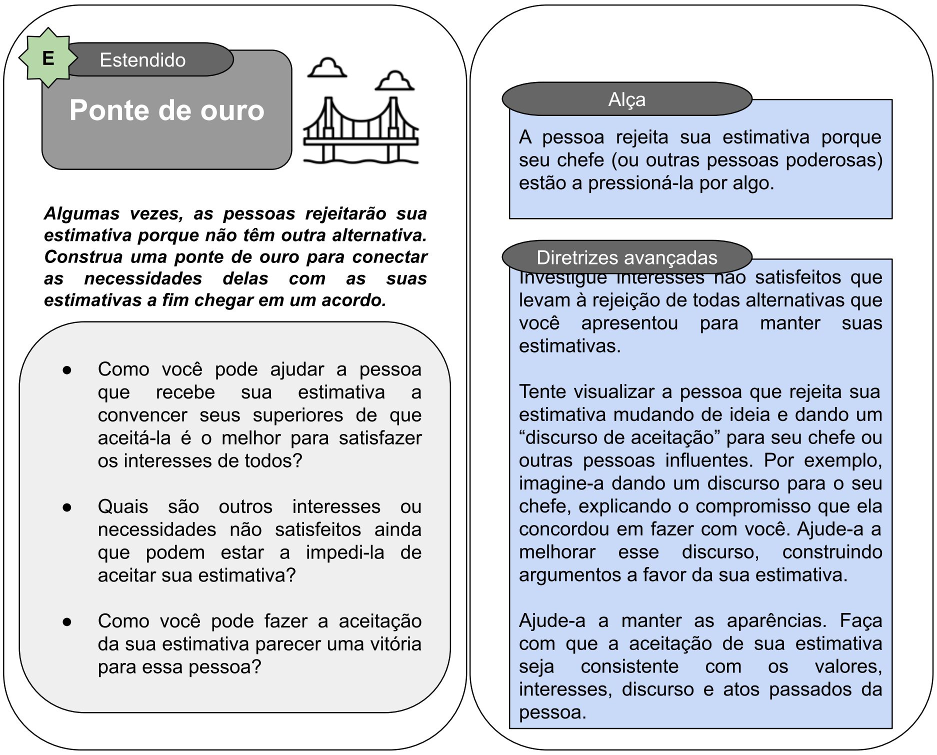


Fig. 10 - Lente Ponte de ouro.

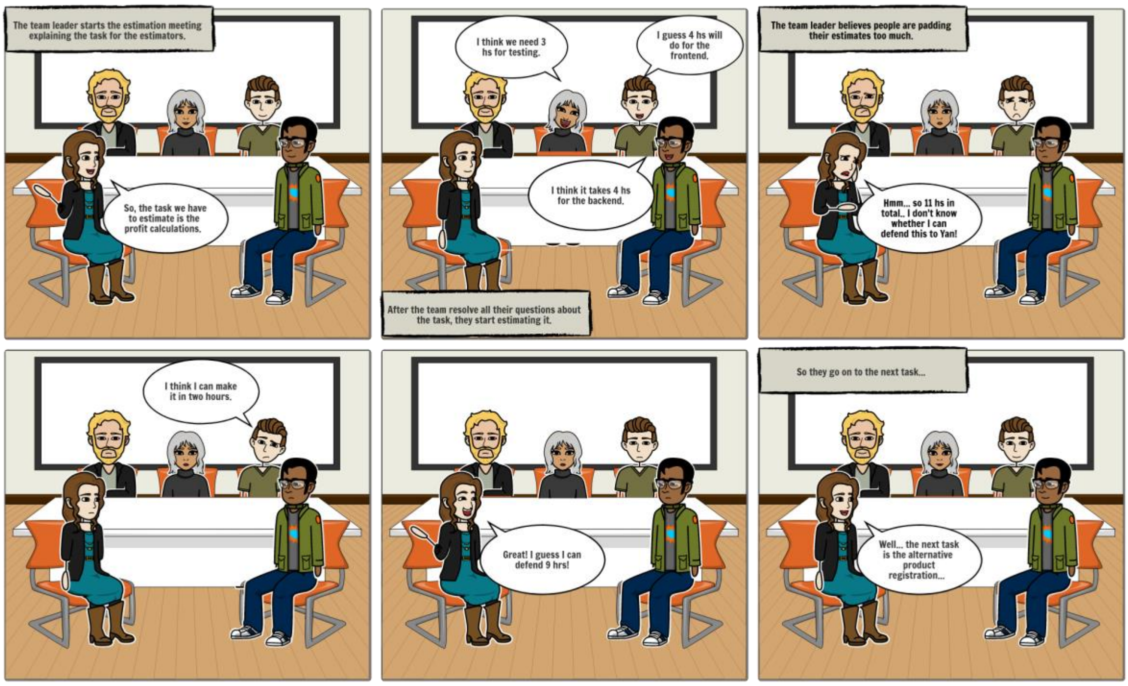
Em relação ao uso em si das cartas, se o estimador sofre com táticas de pressão que podem levá-lo a ceder e mudar a estimativa sem razões legítimas para fazê-lo, a primeira coisa é usar a Lente Ganho de Perspectiva. Em seguida, o uso das demais lentes depende muito da situação, e o estimador precisa consultar as alças para identificar as mais apropriadas. Pode ser o caso de apenas uma das demais lentes ser necessárias ou que todas sejam.

Referências

[1] R. Fisher, W. Ury, and B. Patton, Getting to Yes: Negotiating Agreement Without Giving in, 3rd ed. Penguin Books.
[2] W. Ury, Getting Past No: Negotiating in Difficult Situations, Revised edition. Bantam, 2007.
[3] W. Ury, The Power of A Positive No. Hodder & Stoughton, 2012.
[4] S. McConnell, “Politics, Negotiation, and Problem Solving,” in Software Estimation: Demystifying the Black Art, Redmond: Microsoft Press, 2006, pp. 259–270.

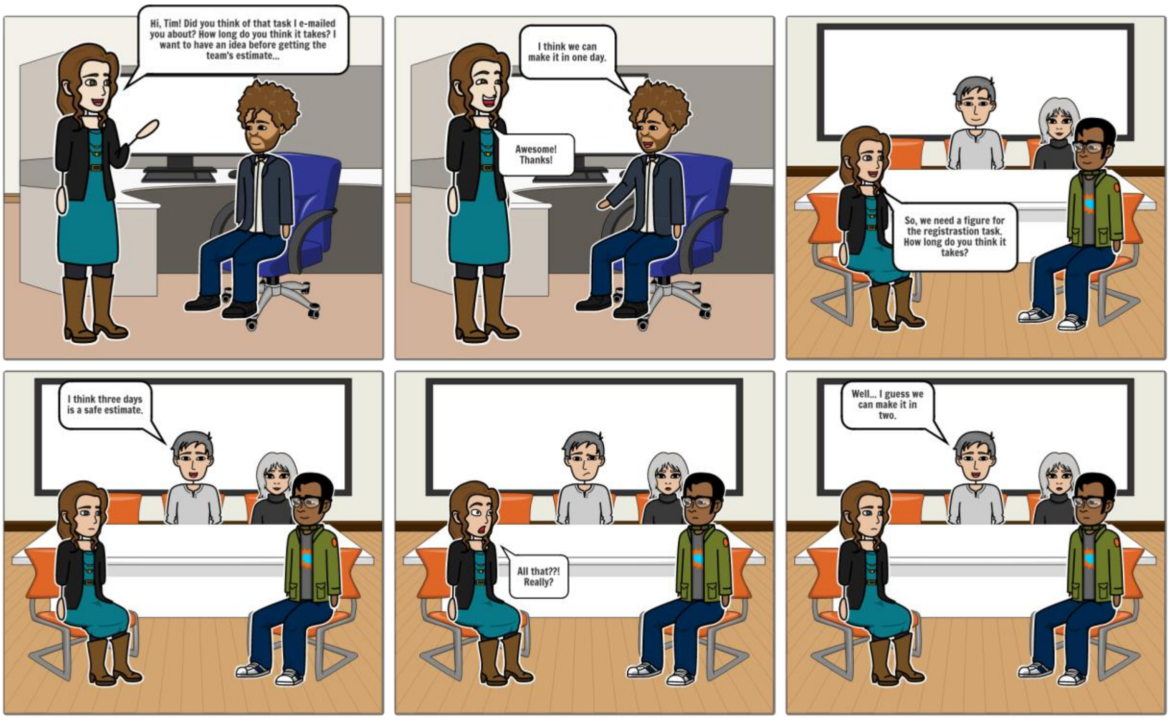
APPENDIX D – SCENARIOS FOR THE FOCUS GROUP

This appendix presents the scenarios that we created as part of the materials for the focus group.



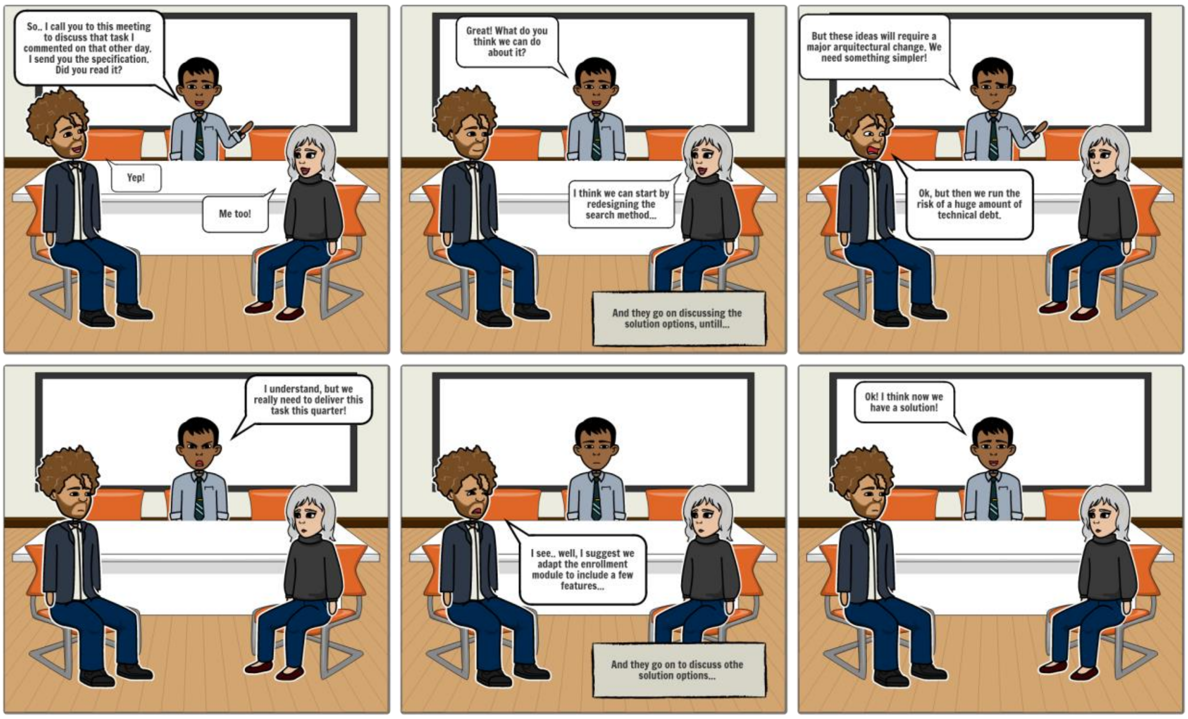
Create your own at Storyboard That

Scenario 1 – Defensible estimates.



Create your own at Storyboard That

Scenario 2 – Pressure makes productive.

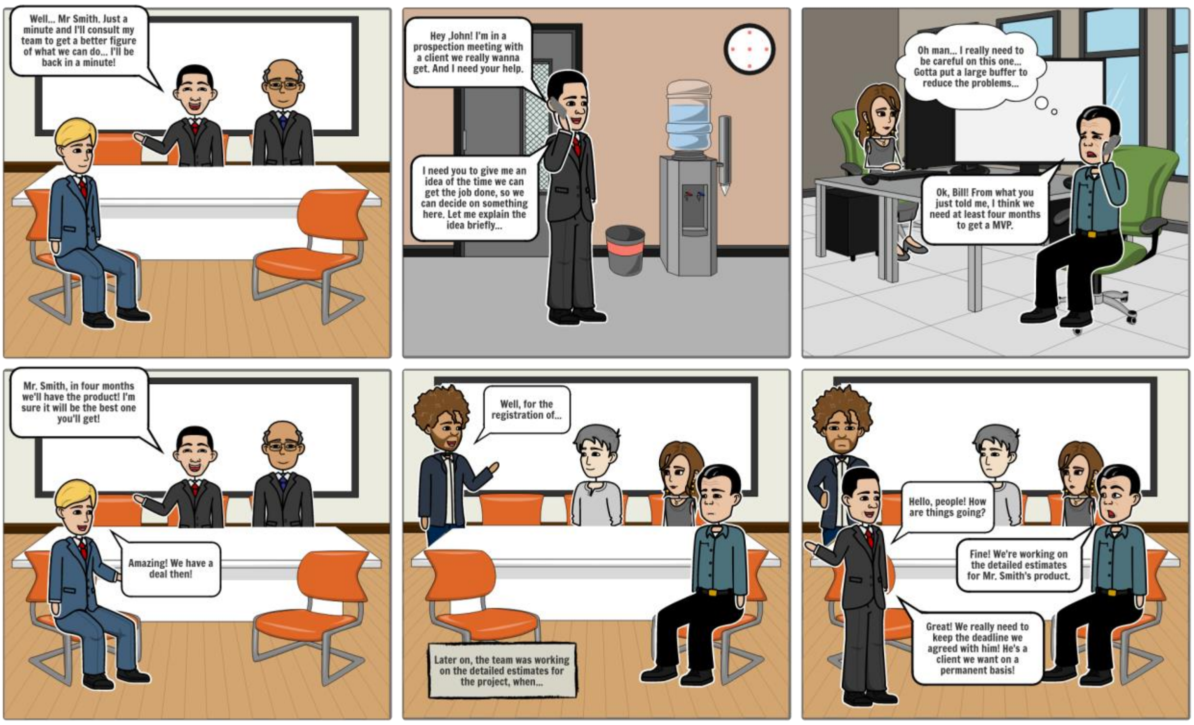


Create your own at Storyboard That

Scenario 3 – Simpler solutions.

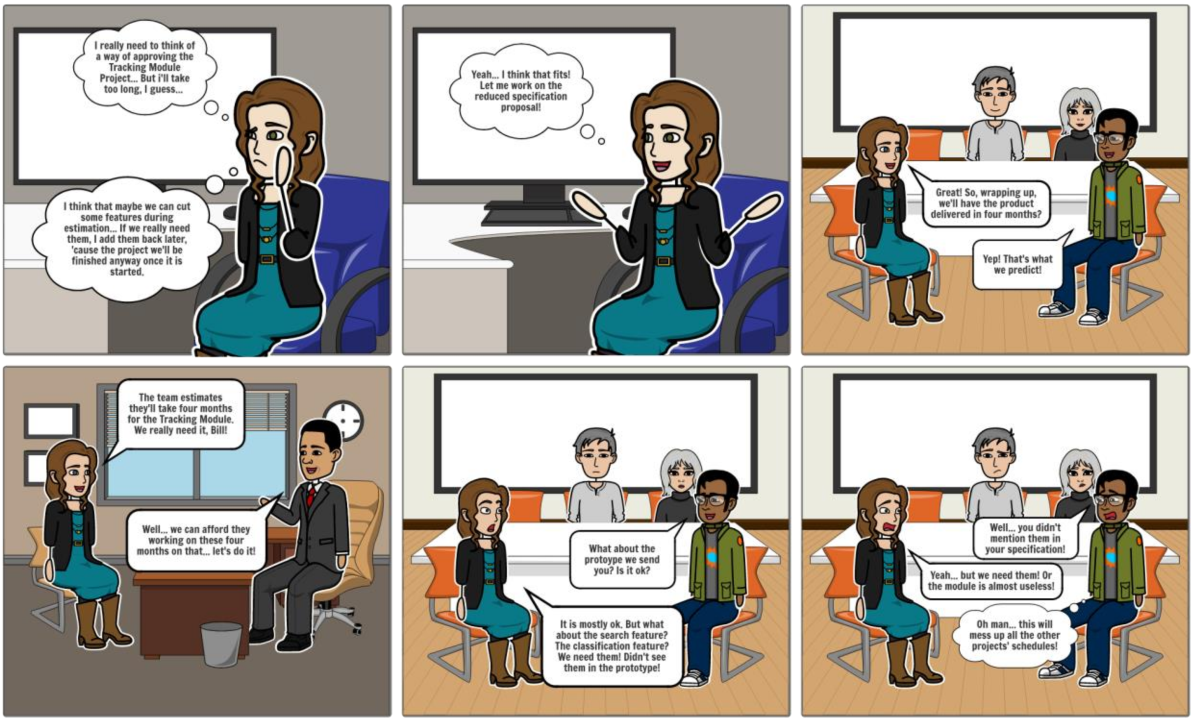
Create your own at Storyboard That

Scenario 4 – The trade show.



Create your own at Storyboard That

Scenario 5 – The ballpark estimate becomes a commitment.

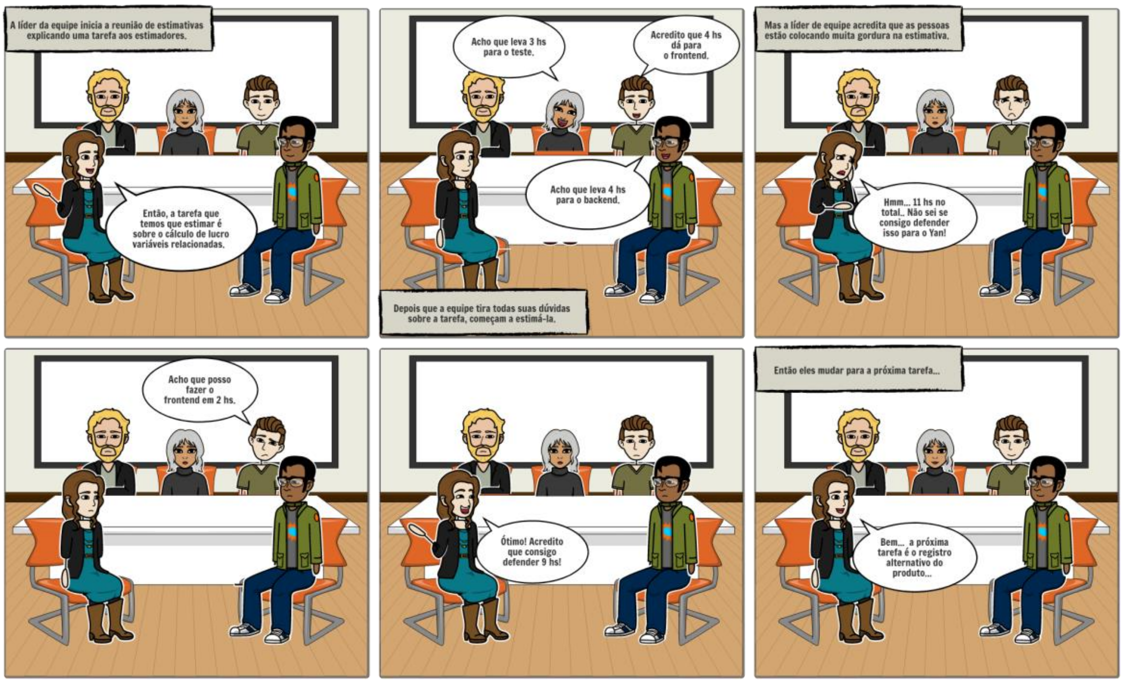


Create your own at Storyboard That

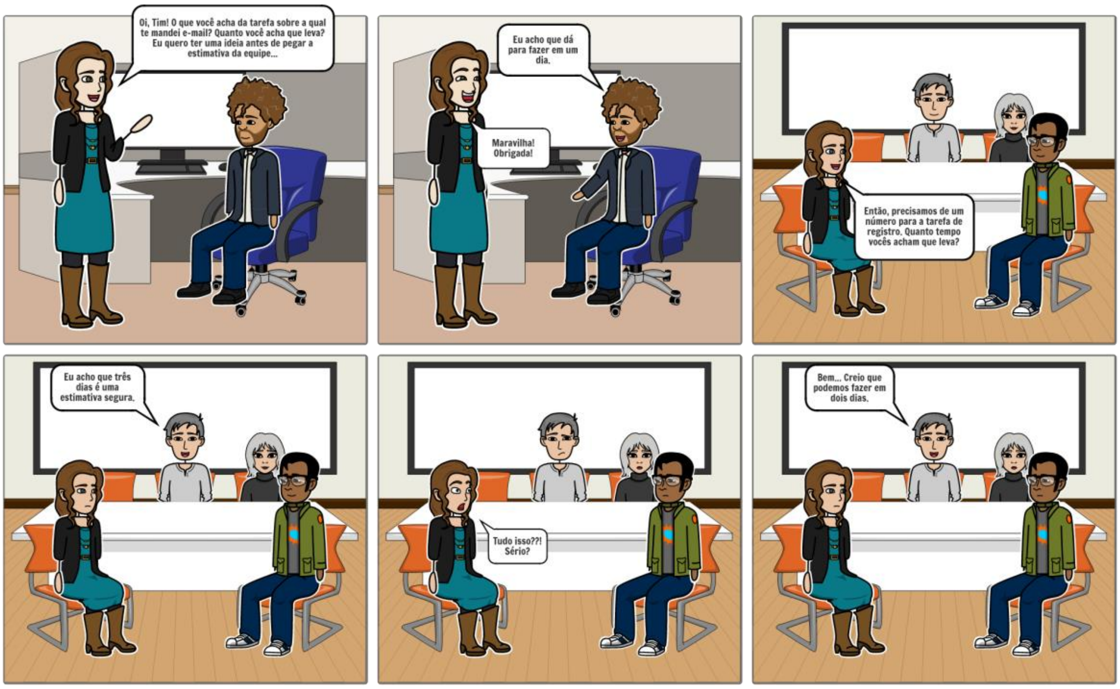
Scenario 6 – Selling ideas.

APPENDIX E – SCENARIOS FOR THE FOCUS GROUP (IN PORTUGUESE)

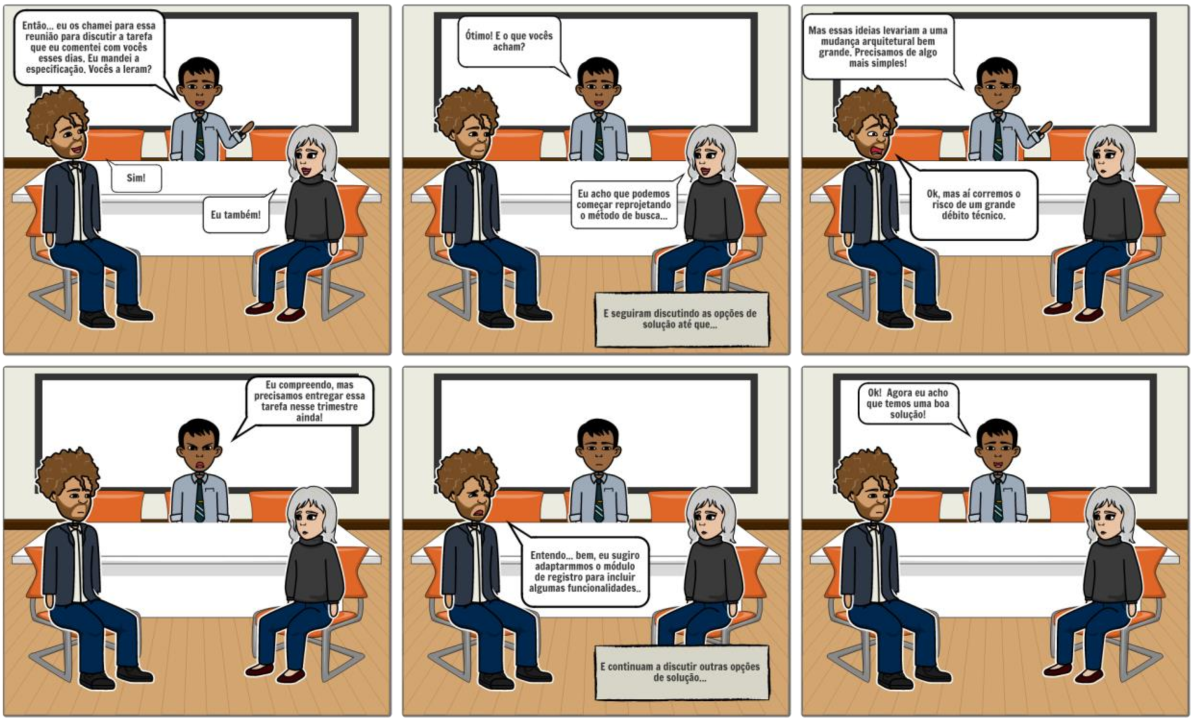
This appendix presents the scenarios that we created as part of the materials for the focus group, translated to Brazilian Portuguese.



Scenario 1 – Defensible estimates.

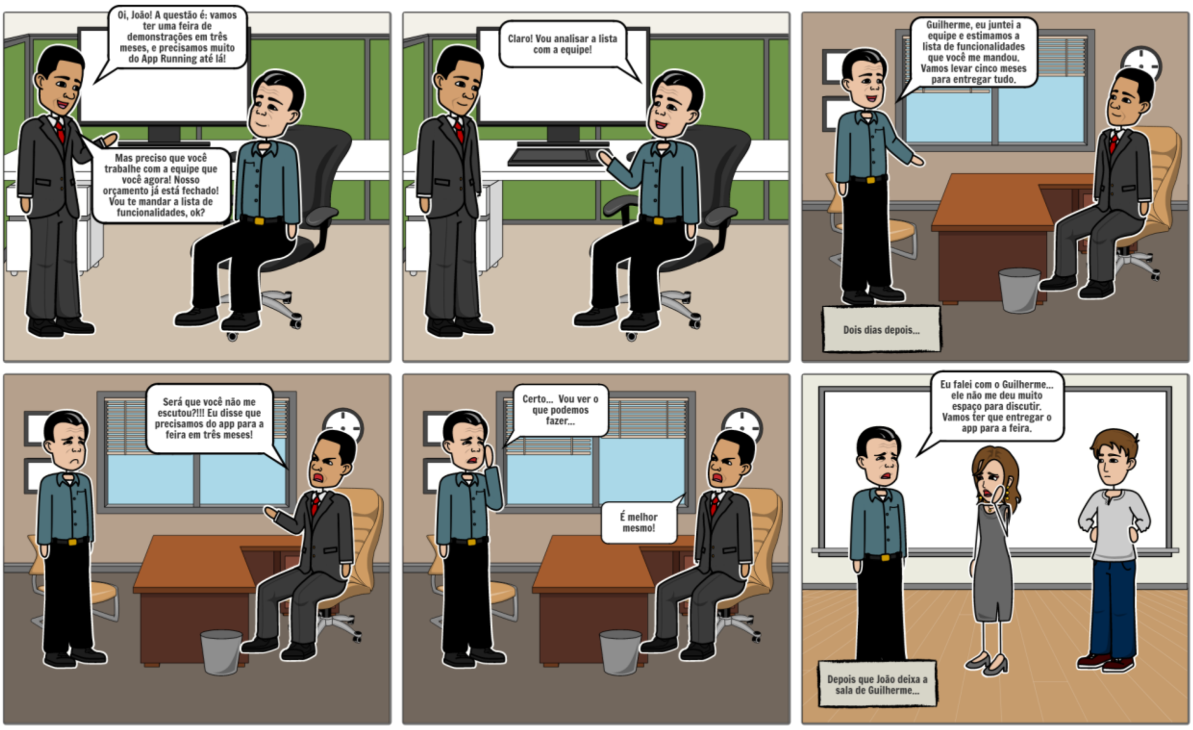


Scenario 2 – Pressure makes productive.



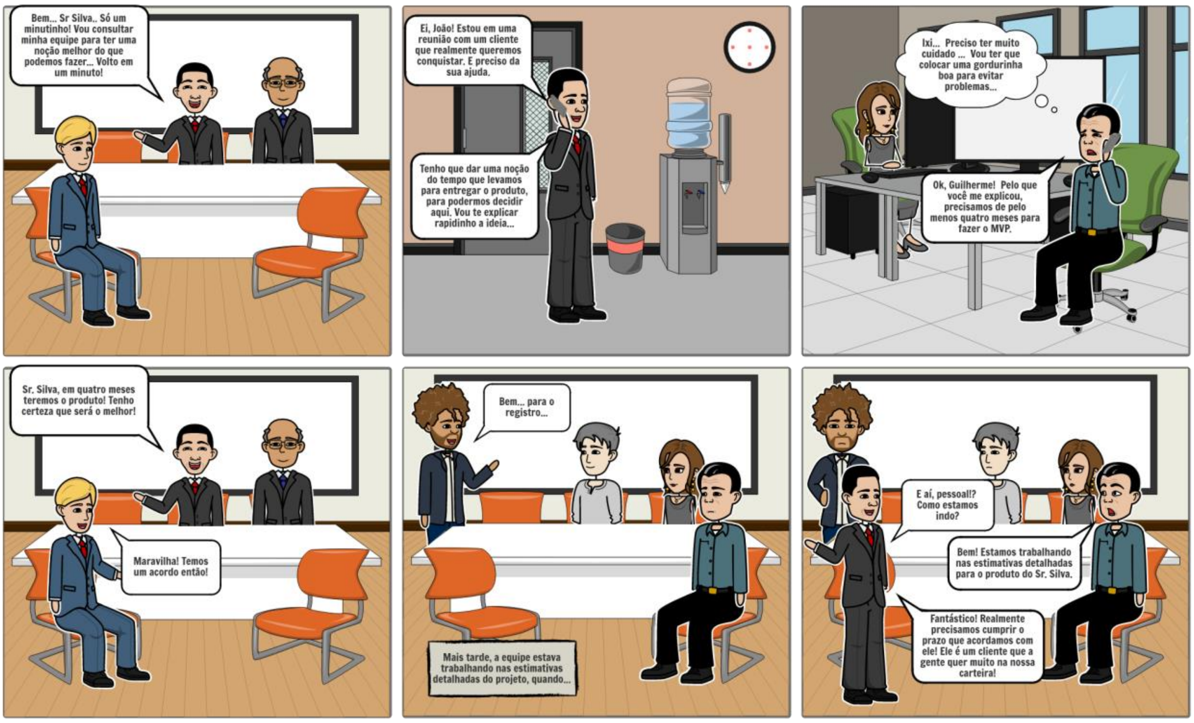
Create your own at Storyboard That

Scenario 3 – Simpler solutions.

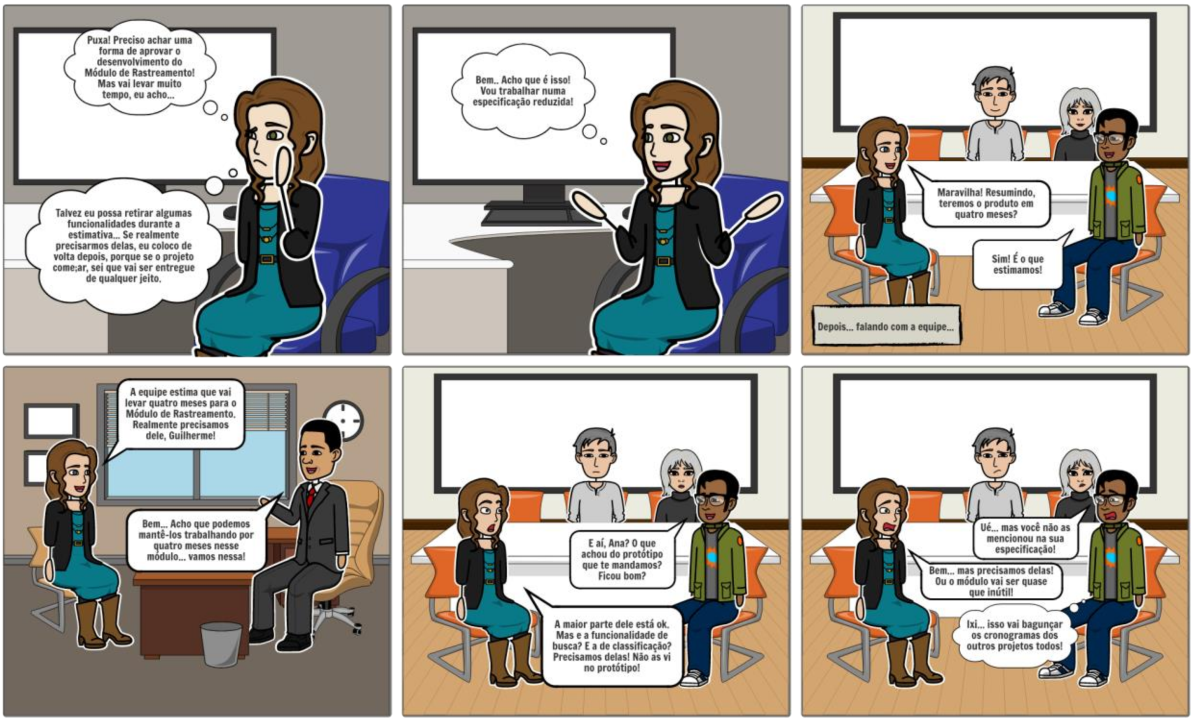


Create your own at Storyboard That

Scenario 4 – The trade show.



Scenario 5 – The ballpark estimate becomes a commitment.



Scenario 6 – Selling ideas.

APPENDIX F –QUESTIONS AND SCENARIOS FOR THE CONTROLLED EXPERIMENT

This appendix presents the questions and scenarios that we created for the controlled experiment.

This appendix presents the questions that are part of the pre- and post-questionnaire mentioned in CHAPTER 7. It includes the demographic questions, Theory of Planned Behavior questions, and scenario questions. **Error! Reference source not found.** presents the demographic questions, which were included only in the pre-questionnaire.

Table 8.9 - Demographic questions.

ID	Question and options
1	What is your age?
	Free text answer
2	What is your gender?
	Woman
	Man
	Prefer not to say
	Other
3	How many years of experience do you have in software development/maintenance?
	Free text answer
4	What roles have you performed in the last year?
	Free text answer
5	What is your highest educational degree?
	Elementary or incomplete high school
	High school
	Incomplete undergraduate
	Graduate
	Incomplete masters
	Masters
	Incomplete Ph.D.
	Ph.D.
6	What is your current responsibility regarding the estimation of software projects or tasks?
	I generate estimates for software development/maintenance projects
	I receive estimates generated by collaborators and communicate them to decision-makers
	I receive estimates generated by collaborators and make decisions based on them

	I have no responsibilities related to software estimates

Table 8.10 shows the reflection questions for the control group in their post-questionnaire.

Table 8.10 - Reflection questions for the control group.

Reminder: we want to know more about pressure over any kind of software estimates: size, effort, duration, or cost. With this in mind, we ask you to answer to the following questions:	
1	**Can you describe situations in which you faced or you witnessed pressure over software estimates in your current or past jobs (including who made pressure, why, and what happened)? In case, you have not lived/witnessed this kind of situation, just inform so.**
	Free text answer
2	**What were the most typical outcomes in such situations (regarding delays, overwork, product quality, or any other aspects you consider relevant)? In case, you have not lived/witnessed this kind of situation, just inform so.**
	Free text answer

Table 8.11 presents the TPB questions. They were included in both questionnaires of the controlled experiment. We dropped the items in red after the reliability analysis.

Table 8.11 - TPB questions.

Estimates are projections about the future. They can be generated by software developers, testers, tech leads, managers, or anyone with the responsibility to provide a value for product size, task effort, or development/maintenance duration and costs. Moreover, we can produce estimates in varied project moments, either individually or in groups.
As the figure below shows, estimates are different from goals and commitments.

In practice, these concepts (estimates, goals, and commitments) can be mixed, creating unreasonable pressure over software estimates.
Considering that, when facing unreasonable pressure during the communication of software estimates or the establishment of commitments, defending estimates involves behaviors such as: - Investigating reasons for pressure, - looking for agreement alternatives, - and trying to keep the estimate realistic or to get around unrealistic commitments, complete the following sentences:

1	Defending an estimate when facing pressure is									**Attitudes**
	harmful	1	2	3	4	5	6	7	beneficial	
2	I am confident that I could defend an estimate when facing pressure if I wanted to.									**Perceived behavioral control**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
3	I expect to defend my software estimates when pressure from stakeholders to change them or to accept an imposed commitment.									**Intention**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
4	It is expected of me that I defend an estimate when facing pressure to change them or to accept an imposed commitment.									**Subjective norms**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
5	Defending an estimate when facing pressure is:									**Attitudes**
	good	1	2	3	4	5	6	7	bad	
6	Most people who are important to me think that									**Subjective norms**
	I should	1	2	3	4	5	6	7	I should not	
	defend an estimate when facing pressure.									
7	The decision to defend an estimate when facing pressure is beyond my control.									**Perceived behavioral control**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
8	I feel intimidated to defend an estimate when facing pressure.									**Subjective norms**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
9	I want to defend my software estimates when facing pressure from stakeholders to change them or to accept an imposed commitment.									**Intention**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
10	Defending an estimate when facing pressure is:									**Attitudes**
	pleasant (for me)	1	2	3	4	5	6	7	unpleasant (for me)	
11	People who are important to me want me to defend an estimate when facing pressure.									**Subjective norms**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
12	Whether I defend an estimate or not is entirely up to me.									**Perceived behavioral control**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
13	I intend to defend my software estimates when facing pressure from stakeholders to change them or to accept an imposed commitment.									**Intention**
	Strongly disagree	1	2	3	4	5	6	7	Strongly agree	
14	For me, to defend an estimate when facing pressure is:									

	Easy	1	2	3	4	5	6	7	Difficult	**Perceived behavioral control**
15	Defending an estimate when facing pressure is:									**Attitudes**
	worthless	1	2	3	4	5	6	7	useful	

Table 8.12 presents the scenario questions included in the post-questionnaire for both the control and the experimental groups.

Table 8.12 - Scenario questions.

Next, we present five scenarios with some kind of pressure over software estimates occurred. After reading the scenario, pick one action alternative that represents how you would react. Observation: there are no right or wrong answers, only the reaction you consider the appropriate one.	
Scenario 1	Suppose that Charles, your boss and one of the company' heads, is in a meeting with Mary, one of the most valuable clients for the company currently. Mary is with a serious inefficiency problem in some of her company manual processes, which she wants to automate. Charles get out of the meeting for a few minutes to call you. After a brief compliment and quick explanation of Mary's situation, he continues: "Mary looked for us regarding an inefficiency problem and I gotta give her some idea of how long it takes to deliver her a product to help, so she can make a decision. I'll quickly explain you the whole idea..." You start to think about the case: you know this estimate will be seen as a commitment not to be broken. So you answer: "We will need four months to release the first version, sir." Charles thanks you and go back to the meeting, closing the deal with Mary with a fixed scope contract. The next day, you already have members from your team talking with business specialists about the process to automate. A few days later, when you are together with your team generating the detailed estimates for this project, Charles enters the room and, after realizing what you are doing, makes a comment: "Do your best, people! We really need to deliver this right. This is a big client, and we cannot disappoint her!" The problem is, after the estimation session, the team believes it takes five to six months to deliver the whole scope. In this situation, what would you do?
a	I turn to the team and remind them: "Have you heard Charles, right? We have to deliver it in four months. Unfortunately, we have no more time than this!”

b	I look for Charles immediately: "I gathered the team to get the detailed estimates for the project and we realized we'll take from four to five months to deliver the first version, but I'll push them a bit to deliver earlier."
c	I look for Charles immediately: "Well... the team thinks we'll take five to six months to deliver the first version. Are you sure that you cannot convince Mary to give us more time on this?”
d	I look for Charles immediately: “We estimate we'll take five to six months to deliver the first product version. However, I believe that if we hire more people, we can deliver it within the initial estimate, considering we did not start it yet. I know this reduces our profit, but at least we keep our image and the team does not suffer so much... what do you think?"
Scenario 2	Suppose your team is in the middle of a Sprint with one month of duration. You still have a few more Sprints before releasing the new system version, but everybody is in a hurry to meet the deadline, which is tight. Moreover, most of the team is not dedicated to the project. So, the product owner calls upon the team. She communicates that, in the last meeting with the client, they realized some necessary changes in a few backlog items, including some you have already implemented in the past and current Sprint. You warn the product owner that, with the changes, you'll need at least one additional Sprint than planned before the release. However, the product owner responds that the contract does not allow for deadline' changes. In this situation, what would you do?
a	I tell the product owner that we cannot make the changes because the contract was sealed and the deadline is tight already.
b	I tell the product owner that if we accept these changes while keeping the deadline, we'll be in trouble.
c	I tell the product owner that I understand that the contract does now allow deadline alterations, but I remind her that the contract also did not mention requirements changes. I ask her then what she thinks will happen if we accept such changes while keeping the deadline.
d	I tell the team we'll work at nights at least for a week in the coming Sprints, and say the pizza is on the house.
Scenario 3	Suppose you work on the software development department of an university. You were called upon to estimate the effort for a coming project. If it gets approved, you'll be part of the development team. You participate of some meetings with the business specialist (a professor that acts as consultor to the dean) for the team to understand the scope and identify requirements. After some days of work with the specialist, the team gets together to estimate

	effort and the delivery deadline. You also consider the other department projects in development and the few human resources you have at your disposal. So, during the last meeting with the specialist, the team presents the estimate and agrees on a final delivery deadline. However, the next day, the specialist calls the team again and says the dean asked for a few more features, saying they are simple things to do. You soon realize that they are not so simple, neither so few. In this situation, what would you do?
a	I leave the meeting with the impression that we will probably not deliver on time.
b	I leave the meeting with the impression that we will do a lot of overtime work in this project.
c	I immediately, in front of the specialist, suggest that the team gets together again to update the estimates and define a new delivery deadline.
d	I ask the specialist whether there are prospects of new hires for our department, seizing the opportunity.
Scenario 4	Suppose John is the technology head in your company, recently promoted (he was the tech lead before that). John invited you and Jane for a meeting to discuss changes to the main company system, that you already talked about in the coffee room. The system has presented performance issues and bugs consistently for the past months, harming the company image. You and Jane discusses a solution already, and were confident that it would take two months to implement it, with two fully dedicated people. However, John reacts immediately: "You're thinking of something that requires a too complicated reestructuring! Can't you just do something simpler and finish in one month maximum? And even in one month, we cannot afford to have two people out of the other projects right now..." In this situation, what would you do?
a	I would get around the situation, saying we would figure out a simpler solution. After all, John was the tech lead and knows a lot about the product. He knows what he is talking about.
b	First of all, I would investigate more about why our proposal was problematic from John's point of view. After all, although John used to be the tech lead, now he is pushed for other reasons.
c	As we were confident on the estimate but we did not want to confront John, I would respond saying that we will try to implement the solution without taking people from other projects, but reinforcing that it probably would not be ready before two months.

d	As we were confident on the estimate, I would reinforce that Jane and I were already trying to spend the minimum of resources and time possible but with less than two month and two people the changes were not possible.
Scenario 5	Suppose that Ana, the team leader, tells you that she registered your next tasks on the system and she also scheduled an estimation session for you two in two days. You read the tasks' descriptions quickly and thinks of some solution ideas. So you note down your estimates for each task. So you remember that Ana has not accepted your estimates so easy lately, always trying to shrink them. This makes you take unrealistic commitments and work under pressure. In this situation, what would you do?
a	I go back to the other tasks I was working on because I have no more time to prepare for the estimation session.
b	I readily draft the subtasks of the tasks I am estimating, justifying my estimates more robustly. I also prepare to discuss what can go wrong if we reduce these estimates.
c	I pad the tasks' estimates because Ana is trying to reduce them anyway.
d	I look for Ana before the meeting to ask whether she can give some of these tasks to someone else because I am overloaded already. So, if she tries to reduce the estimates, at least I will not have too many tasks.

Table 8.13 shows the questions about perceived usefulness included in the post-questionnaire of the experimental group only. They were positioned after the scenario questions.

Table 8.13 - Perceived usefulness questions for the experimental group.

1	**Do you think that learning negotiation principles like the ones presented in the interactive videos is useful for the *defense of software estimates* in your current work environment?**
	Yes
	No
	I don't know
2	**If yes: Explain which lenses or principles you believe are the most useful for the defense of software estimates in your current job (and why).**
	Free text answer
3	**Otherwise (No or I don't know): For what reason(s)?**
	Free text answer

APPENDIX G – ADDITIONAL ANALYSIS OF THE CONTROLLED EXPERIMENT

This appendix presents the reliability analysis, the between groups analysis before the intervention, and within the control group.

This appendix presents complementary analysis to the ones that CHAPTER 7 describes. It starts with the reliability analysis. Next, it brings the between group analysis for data before the interventions. Finally, it describes the analysis of data within the control group.

1. Reliability Analysis

We start with a reliability analysis of the questionnaires constructed for collecting data regarding the Theory of Planned Behavior (TPB). For each relevant variable, we calculated Cronbach's α for the data we had at the first moment of data collection when we had 45 participants. We present the results in Figures 1-4 for attitudes, subjective norms, perceived behavioral control, and intentions, respectively.

Figure 1 - Attitudes' reliability scores for the 1st moment of data collection.

Attitudes ▾

Frequentist Scale Reliability Statistics

Estimate	Cronbach's α	mean	sd
Point estimate	0.658	19.156	5.334
95% CI lower bound	0.450	17.597	4.416
95% CI upper bound	0.797	20.714	6.738

Frequentist Individual Item Reliability Statistics

	If item dropped			
Item	Cronbach's α	Item-rest correlation	mean	sd
M1TPB1	0.541	0.517	5.000	1.784
M1TPB5	0.492	0.570	4.733	1.993
M1TPB15	0.568	0.480	5.511	1.727
M1TPB10	0.735	0.234	3.911	2.065

Note. The following items were reverse scaled: M1TPB5, M1TPB10.

Figure 2 - Subjective norms' reliability scores for the 1st moment of data collection.

Subjective Norms ▼

Frequentist Scale Reliability Statistics

Estimate	Cronbach's α	mean	sd
Point estimate	0.703	18.800	5.446
95% CI lower bound	0.527	17.209	4.509
95% CI upper bound	0.822	20.391	6.880

Frequentist Individual Item Reliability Statistics

	If item dropped			
Item	Cronbach's α	Item-rest correlation	mean	sd
M1TPB4	0.538	0.641	5.089	1.905
M1TPB6	0.621	0.518	5.111	2.036
M1TPB8	0.774	0.238	3.444	1.765
M1TPB11	0.579	0.590	5.156	1.770

Note. The following item was reverse scaled: M1TPB6.

Figure 3 - Perceived behavioral control' reliability scores for the 1st moment of data collection.

Perceived Behavioral Control ▼

Frequentist Scale Reliability Statistics

Estimate	Cronbach's α	mean	sd
Point estimate	0.334	19.178	3.810
95% CI lower bound	−0.079	18.065	3.154
95% CI upper bound	0.607	20.291	4.812

Note. The following items correlated negatively with the scale: M1TPB7, M1TPB12.

Frequentist Individual Item Reliability Statistics

	If item dropped			
Item	Cronbach's α	Item-rest correlation	mean	sd
M1TPB7	0.203	0.232	5.156	1.651
M1TPB12	0.318	0.144	4.644	1.885
M1TPB2	0.145	0.310	5.511	1.359
M1TPB14	0.409	0.053	3.867	1.660

Note. The following items were reverse scaled: M1TPB7, M1TPB14.

Figure 4 - Intentions' reliability scores for the 1st moment of data collection.

Intentions ▾

Frequentist Scale Reliability Statistics

Estimate	Cronbach's α	mean	sd
Point estimate	0.818	16.044	4.226
95% CI lower bound	0.698	14.810	3.499
95% CI upper bound	0.895	17.279	5.339

Frequentist Individual Item Reliability Statistics

	If item dropped			
Item	Cronbach's α	Item-rest correlation	mean	sd
M1TPB3	0.673	0.754	5.467	1.531
M1TPB9	0.832	0.616	5.022	1.889
M1TPB13	0.753	0.675	5.556	1.486

The Cronbach's α reached acceptable levels (above 0.7) for subjective norms and intentions. Analysis of dropping items suggests we can increase Cronhach's α to over 0.7 for attitudes by dropping item M1TPB10. We also increase Cronhach's α for subjective norms when dropping item M1TPB 8 and perceived behavioral control when dropping item M1TPB14. However, for the latter, we do not reach an acceptable level for Cronhach's α. This was probably because we did not have enough items to measure it. At this point, a possibility was to write down more items. However, this would require participants to spend more time answering the questionnaire, and we would need another step of data collection with the new questionnaire to make answers before and after the interventions comparable. Thus, we decided to proceed with the original questionnaire to avoid increasing mortality among participants, dropping the abovementioned items to raise reliability.

2. Differences Before the Intervention

This section analyzes the differences between the control and experimental group at the first moment of data collection regarding TPB variables, i.e., before any intervention. Although CHAPTER 7 presents the boxplots, Figure 5 presents the descriptive statistics. In this figure, Group 1 represents the experimental group, and Group 2 represents the control one.

Figure 5 - Descriptive statistics for each group in the 1st moment of data collection.

Descriptive Statistics ▼

	M1A		M1SN		M1PBC		M1I	
	1	2	1	2	1	2	1	2
Valid	16	16	16	16	16	16	16	16
Missing	0	0	0	0	0	0	0	0
Median	14.000	14.500	12.000	17.500	16.000	16.500	16.000	18.000
Mean	13.875	15.250	12.188	17.125	15.125	15.438	14.563	17.313
Std. Deviation	4.801	5.053	4.996	4.048	3.324	3.669	4.335	4.238
Minimum	6.000	3.000	3.000	6.000	9.000	9.000	4.000	6.000
Maximum	21.000	21.000	19.000	21.000	20.000	21.000	19.000	21.000

We also present the boxplots in Figures 6-9 for each TPB variable.

Figure 6 - Boxplot for attitudes at the 1st moment of data collection.

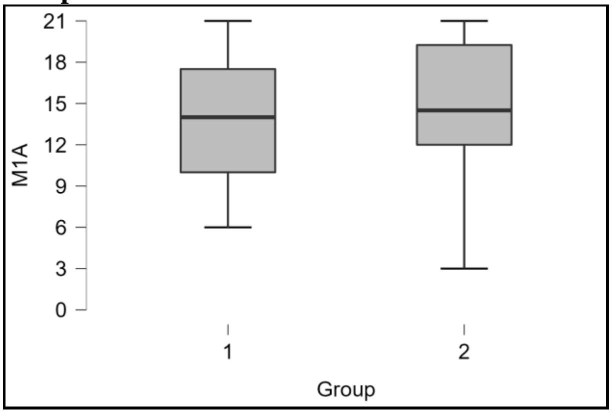


Figure 7 - Boxplot for subjective norms at the 1st moment of data collection.

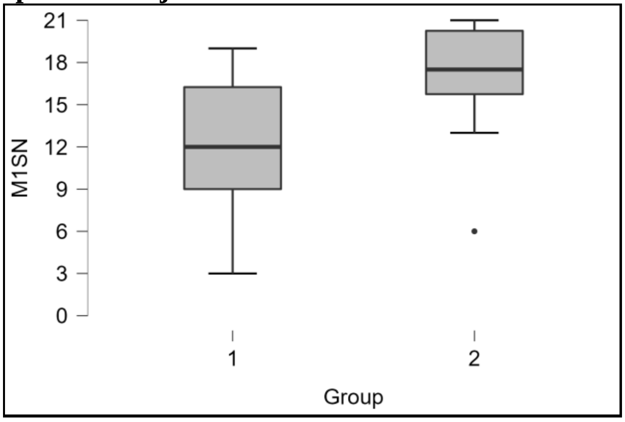


Figure 8 - Boxplot for perceived behavioral control at the 1st moment of data collection.

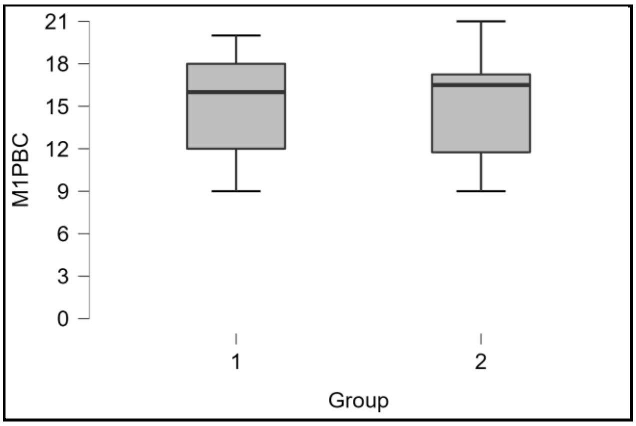
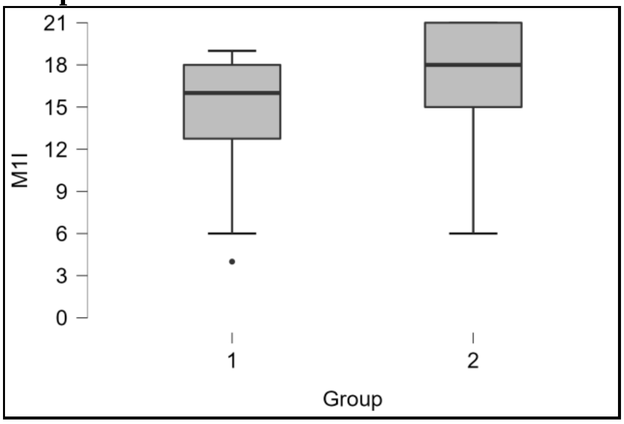


Figure 9 - Boxplot for intentions at the 1st moment of data collection.



Visual inspection of the data suggests the control group has a higher score for subjective norms than the experimental. There also seem to have a slightly higher score for intentions. Testing the differences using Bayesian Hypothesis Testing (with a null hypothesis of no differences and a Cauchy prior with σ = 0.707), Figure 10 shows moderate evidence in favor of the alternative hypothesis as a better explanation of the data than the null one regarding subjective norms, due to a Bayes-Factor above 3. We cannot say the same from other variables, although we only have weak evidence for the null hypothesis regarding intentions.

Figure 10 - Bayesian Hypothesis Testing at the 1st moment of data collection.

Bayesian Mann-Whitney U Test

	BF_{10}	W	Rhat
M1A	0.434	108.000	1.000
M1SN	5.010	54.500	1.000
M1PBC	0.357	124.000	1.000
M1I	1.390	79.000	1.000
Age	0.342	124.000	1.000
Exp	0.395	114.500	

Note. Result based on data augmentation algorithm with 5 chains of 10000 iterations.

Figure 10 also shows no relevant differences between the groups regarding age and years of experience with software development. In Figure 11, we present the equivalent Null Hypothesis Significance Testing, showing significant results only for subjective norms. We also present the test of normality of variables (using Shapiro-Wilk), revealing non-normal distributions for subjective norms (for the control group) and intentions (for both groups).

Figure 11 - Null Hypothesis Significance Testing at the 1st moment of data collection.

Independent Samples T-Test ▼

	W	df	p	Rank-Biserial Correlation
M1A	108.000		0.460	−0.156
M1SN	54.500		0.006	−0.574
M1PBC	124.000		0.894	−0.031
M1I	79.000		0.066	−0.383
Age	124.000		0.895	−0.031
Exp	114.500		0.622	−0.105

Note. For the Mann-Whitney test, effect size is given by the rank biserial correlation.
Note. Mann-Whitney U test.

Assumption Checks

Test of Normality (Shapiro-Wilk)

		W	p
M1A	1	0.946	0.424
	2	0.889	0.054
M1SN	1	0.949	0.480
	2	0.857	0.017
M1PBC	1	0.914	0.136
	2	0.932	0.259
M1I	1	0.823	0.006
	2	0.825	0.006
Age	1	0.921	0.173
	2	0.905	0.097
Exp	1	0.892	0.061
	2	0.887	0.050

Note. Significant results suggest a deviation from normality.

3. Within Group Analysis – Control Group

In CHAPTER 7, we mention that we tested for the differences before and after the reflection questions for the control group. In Figure 18, we present the Bayesian Hypothesis Testing results.

Figure 18 - Bayesian Hypothesis Testing within the control group.

Bayesian Wilcoxon Signed-Rank Test					
Measure 1		Measure 2	BF_{+0}	W	Rhat
M2A	-	M1A	2.403	59.500	1.000
M2SN	-	M1SN	0.203	39.500	1.000
M2PBC	-	M1PBC	0.192	28.000	1.000
M2I	-	M1I	0.329	23.500	1.000

Note. For all tests, the alternative hypothesis specifies that Measure 1 is greater than Measure 2. For example, M2A is greater than M1A.
Note. Result based on data augmentation algorithm with 5 chains of 10000 iterations.

Figures 19-22 present the graphs showing effect sizes' media, confidence intervals, and distributions of priors and posteriors.

Figure 19 - Test of difference of attitudes within the control group.

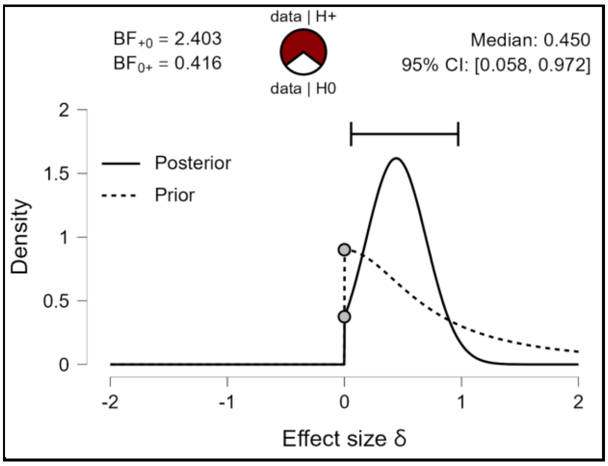


Figure 20 - Test of difference of subjective norms within the control group.

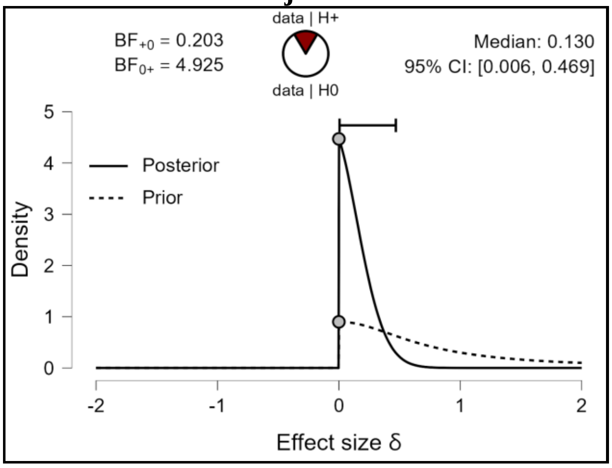


Figure 21 - Test of difference of perceived behavioral control within the control group.

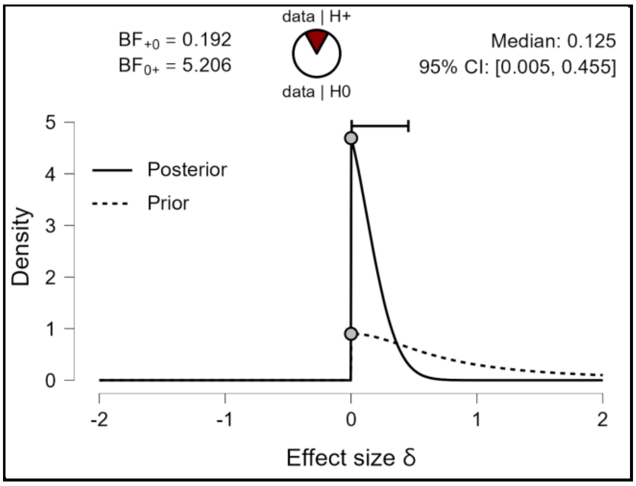
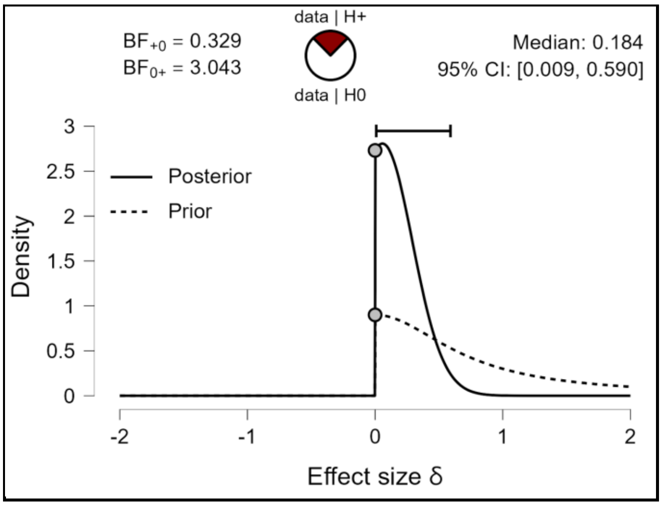


Figure 22 - Test of difference of intentions within the control group.



4. Qualitative Analysis

This section presents the qualitative analysis regarding the perceived usefulness of the defense lenses by the experimental group' participants and the answers to the reflection questions by the control group' participants. Table 8.14 presents examples of supporting quotes for each of the lens that participants mentioned in their answers about the perceived usefulness of the lenses.

Table 8.14 - Perceived usefulness of the defense lenses.

Lens	Supporting quotes	Participant
Assert your estimate	"Assertion [of your estimate], because it exercises the appreciation of the rationale that lies behind the estimate"	P5
Laddering whys	"Asking questions makes us reflect on motivations and difficulties of people involved in the projects, and I think this will be useful in my estimates."	P18
Pressuring forces	"Overall, the explanations are about negotiation tactics. This way, we must establish a communication channel where, on one side, we show	P10

	that our estimate is what it is because of n tasks and m subtasks. On the other side, we must try to understand the reasons for pressure. So, we can get to a beneficiall path for both sides and focus on delivering value."	
Candidade commitments	"In my experience, flexibility is really important in negotiations. Trying to find a commitment that brings benefits to all parties is hard, but I believe is the best alternative."	P29
Choose your battles	Noone mentioned it. So, there is no quotation.	-
Keep strategy	"Because I always have reasons outside my control, so I can look for tools to defend my estimate."	P2
Perspective taking	"Perspective Taking, because when we take a wider look we can have arguments to base or to balance a point of view."	P11
Reality test Golden bridge	"I specially enjoyed the last two lenses. The one that explains that asking what is going to happen is better than telling (Reality test), and the one about making the acceptance of the estimate to look like a win (Golden bridge)."	P4
All lenses	"I think the principles were interesting. It is possible to balance time x quality. I believe the lenses complement each other, and knowing how to negotiate is part of the job"	P35

Table 8.15 presents supporting quotations for the codes that we created to reflect the outcomes of pressure mentioned by the participants of the control group in their answers.

Table 8.15 - Reflection questions.

Outcome of pressure codes	Supporting quotations	Participant
Increase in product failure/bugs	"The product can fail in critical moments for the users causing damages to trust that can take more time to recover than any change in engineering."	P12
Product instability	"For me, the most typical outcome is the lack of software stability in the sense that we do not know what the impact of changes over other functionalities are. To make such a large change in a software with no tests is like a Pandora box. We will experience delays, strong pressure, low quality for the new product version, software instability."	P8
Neglect of long-run maintenance Neglect of testing activities	"Estimates made under pressure tend to neglect the project maintenance in the long-run. One of the first parts to be neglected is test coverage documentation generating problems when revisiting the problem or when new people join the project."	P12
Neglect of good practices	"The product quality was affected, because the first thing to cut was automated tests, and next the technical team started to skip the good practice and standards to save time."	P27
Overall lower quality	"The most typical outcome is lost in product quality."	P6
Unmet expectations	"Lack of accuracy in planning the next sprints. When the situation is recurrent the problems come back on unexpected moments. So, we have higher chances of a sprint's expectations are not met."	P12

Unmet needs	"Low quality products and that do not satisfy the client's needs."	P23
Overtime work	"Sometimes, quality is left behind so that we can have a fast solution, and when we have a good quality fast solution it requires over time work."	P1
Emotional distress	"Personal damage. Team damage. Negative team entropy. Low professional steem for everyone involved."	P20
Resignation	Talking about a large change in a legacy product with no tests required by superiors because they did not want to spend money on third party software: "We had two months to modify the flows to support the new requirements. It was extremely stressful, and we obviously did not deliver the complete changes. In the end of February, I had burnout and resigned. I was a developer, had no voice to negotiate."	P8
Solution-block	"When a delay happens, sometimes there is overtime work, and the developer might not devise a solution because they are trying to solve the problem for so many hours already. Sometimes the code is of low quality and even impacts testing."	P19

ANNEX A – BIG-FIVE 20 ITEMS

This annex presents the Big-Five 20-items questionnaire in Brazilian Portuguese [228].

INSTRUÇÕES. A seguir são apresentadas 20 afirmações que tratam de características pessoais. Leia cada uma com atenção e, utilizando a escala de resposta abaixo, indique o quanto concorda ou discorda com o fato de cada característica descrevê-lo.

1	2	3	4	5
Discordo totalmente	Discordo em parte	Nem concordo nem discordo	Concordo em parte	Concordo totalmente

Eu me vejo como alguém que...

01.____É conversador, comunicativo.
02.____É minucioso, detalhista no trabalho.
03.____Insiste até concluir a tarefa ou o trabalho.
04.____Gosta de cooperar com os outros.
05.____É original, tem sempre novas idéias.
06.____É temperamental, muda de humor facilmente.
07.____É inventivo, criativo.
08.____É prestativo e ajuda os outros.
09.____É amável, tem consideração pelos outros.
10.____Faz as coisas com eficiência.
11.____É sociável, extrovertido.
12.____É cheio de energia.
13.____É um trabalhador de confiança.
14.____Tem uma imaginação fértil.
15.____Fica tenso com frequência.
16.____Fica nervoso facilmente.
17.____Gera muito entusiasmo.
18.____Gosta de refletir, brincar com as idéias.
19.____Tem capacidade de perdoar, perdoa fácil.
20.____Preocupa-se muito com tudo.

Itens por fator

Abertura à experiência
Itens 05, 07, 14 e 18.

Conscienciosidade
Itens 02, 03, 10 e 13.

Extroversão
Itens 01, 11, 12 e 17.

Amabilidade
Itens 04, 08, 09 e 19.

Neuroticismo
Itens 06, 15, 16 e 20.
.

ANNEX B – ASSERTIVENESS SCALE

This annex presents the assertiveness scale in Brazilian Portuguese [229].

ESCALA DE ASSERTIVIDADE RATHUS – RAS

01. A maioria das pessoas parece ser mais agressiva e assertiva do que eu.

02. Eu tenho hesitado em marcar ou aceitar encontros por causa de minha "timidez".

03. Quando a comida servida em um restaurante não é do meu agrado, eu reclamo ao garçom ou garçonete.

04. Eu tomo cuidado para evitar magoar os sentimentos das pessoas, mesmo quando sinto que fui ofendido.

05. Se um vendedor faz grande esforço para me mostrar mercadoria que não é exatamente o que eu queria, tenho dificuldade em dizer "Não".

06. Quando me pedem para fazer alguma coisa, eu insisto em saber o por quê.

07. Existem momentos em que gosto de uma boa "briguinha".

08. Eu procuro progredir na vida tanto quanto a maioria das pessoas em minha posição profissional.

09. Para dizer a verdade, as pessoas frequentemente tiram vantagem de mim.

10 Gosto de iniciar conversa com pessoas que acabo de conhecer e com estranhos.

11. Frequentemente não sei o que dizer a pessoas atraentes do sexo oposto.

12. Eu hesitaria em fazer chamadas telefônicas para estabelecimentos comerciais e instituições.

13. Eu preferiria escrever uma carta para pedir emprego ou admissão em uma instituição do que submeter-se a uma entrevista cara-a-cara.

14. Eu acho embaraçoso devolver mercadorias defeituosas.

15. Se um parente próximo e respeitado estiver me aborrecendo, prefiro abafar meus sentimentos do que expressar meu aborrecimento.

16. Tenho evitado fazer perguntas por receio de parecer ignorante (burro).

17. Às vezes, durante uma discussão, tenho receio de ficar tão aborrecido (transtornado) e começar a tremer todo.

18. Se um conferencista famoso e respeitado faz uma declaração que penso estar incorreta, farei com que meu ponto de vista seja igualmente ouvido.

19. Eu evito discutir preços com balconistas e vendedores.

20. Quando faço alguma coisa importante ou que vale a pena, eu dou um jeito para que as outras pessoas fiquem sabendo.

21. Sou aberto e franco sobre os meus sentimentos.

22. Se alguém vem espalhando estórias falsas a meu respeito, eu o procuro o mais rápido possível para termos uma conversa sobre o assunto.

23. Eu frequentemente tenho dificuldade em dizer "Não".

24. Eu tendo a reprimir minhas emoções ao invés de fazer uma cena (um escândalo).

25. Eu reclamo do serviço quando o acho deficiente em um restaurante ou qualquer lugar.

26. Quando recebo um elogio, às vezes não sei o que dizer.

27. Se um casal perto de mim em um teatro ou em uma conferência estiver conversando alto, eu pediria para ficarem quietos ou para irem conversar em outro lugar.

28. Se alguém fura a fila na minha frente, está me provocando para briga.

29. Sou rápido para expressar uma opinião.

30. Existem momentos quando não consigo dizer coisa alguma.