The best defense is a good defense: adapting negotiation methods for tackling pressure over software project estimates

Patricia G. F. Matsubara
patriciagfm@icomp.ufam.edu.br
Federal University of Amazonas (UFAM) & Federal
University of Mato Grosso do Sul (UFMS)
Manaus-AM & Campo Grande-MS, Brazil

Igor Steinmacher
igorfs@utfpr.edu.br
Federal University of Technology - Paraná (UTFPR)
Campo Mourão-PR, Brazil

Bruno Gadelha
bruno@icomp.ufam.edu.br
Federal University of Amazonas (UFAM)
Manaus-AM, Brazil

Tayana Conte
tayana@icomp.ufam.edu.br
Federal University of Amazonas (UFAM)
Manaus-AM, Brazil

ABSTRACT

Software estimation is critical for a software project's success and a challenging activity. We argue that estimation problems are not restricted to the generation of estimates but also their use for commitment establishment: project stakeholders pressure estimators to change their estimates or to accept unrealistic commitments to attain business goals. In this study, we employed a Design Science Research (DSR) methodology to design an artifact based on negotiation methods, to empower software estimators in defending their estimates and searching for alternatives to unrealistic commitments when facing pressure. The artifact is a concrete step towards disseminating the soft skill of negotiation among practitioners. We present the preliminary results from a focus group that showed that practitioners from the software industry could use the artifact in a concrete scenario when estimating and establishing commitments about a software project. Our future steps include improving the artifact with the suggestions from focus group participants and evaluating it empirically in real software projects in the industry.

CCS CONCEPTS

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

KEYWORDS

Software effort estimation, commitment, negotiation

1 INTRODUCTION

Estimating software projects is a critical activity in software engineering. Software practitioners (here referred to as estimators) produce estimates about at least one parameter: size, effort, cost, or duration [21]. Extensive research investigated the factors affecting estimators, and the generation of software estimates, including reasons for errors and strategies to improve accuracy [2, 13].

We argue that the generation of estimates is not the only problem: the use of software estimates for commitments establishment in software projects also plays a crucial role in accuracy. Our argument builds on the fact that other project stakeholders, like upper management and clients (referred to as receptors), receive these estimates. Moreover, estimators and receptors must agree on commitments: promises to deliver a set of features on a deadline, at a specific level of quality [22]. Ideally, the technical estimates from estimators are the foundations to define such commitments. Nevertheless, ultimately, people develop and maintain software to satisfy desirable business outcomes—or targets [22]. Usually, receptors want to hit these targets. Previous research has shown that sometimes estimators are pressured to change their estimates [17] to meet targets when the software estimates collide with them [5], making estimates acceptable to receptors, or "defensible" [18]. In addition, time pressure—the perception that time is scarce considering the tasks' demands—is standard in the software industry, and commercial pressure is behind many of its causes [16]. Time pressure can increase efficiency up to a certain point and on the cost of an efficiency-quality trade-off [16]. This reverberates in surveys in the software industry, where tight deadlines were the most cited and most likely cause for technical debt [26, 28]. In summary, when estimators fail to defend their estimates when facing pressure, consequences go beyond inaccurate estimates: software quality also suffers.

Unfortunately, software practitioners do not possess the skills needed to truly defend their estimates from pressure [20]. In this context, we aim to answer the following research question: **How to empower software practitioners to resist pressure and defend their software estimates during commitment establishment?** In the next section, we argue for the adaptation of negotiation methods to the estimation context to answer this research question. This endeavor is innovative: it is a concrete step to enable software practitioners to develop the relevant soft skill of negotiation. As

such, our work is a response to the call to promote soft skills in Software Engineering [3], contributing to close the gaps regarding what the software industry needs from practitioners [10], and to address essential soft skills that complement technical skills [4].

2 BACKGROUND

According to Schneider et al. [29], negotiation encompasses any situation that involves two or more parties that (i) consider the situation a negotiation and prepare accordingly; (ii) try to gain something with the interaction or to improve their situation; (iii) need each other to get their purpose; (iv) communicate back-and-forth, questioning each other, sharing information, and making offers or creating options; and (v) work towards an agreement over a result or a behavior change. We argue that the interaction of the estimation process and commitment establishment in software projects presents many of the elements of this definition. For instance, when estimating, people prepare to negotiate, with some stakeholders increasing the estimates while others do the opposite [17].

Next, we summarize the three negotiation methods that form our theoretical foundations. We start with principled negotiation [9]: a groundbreaking method [25]. It shifted the approach from bargaining to a more flexible one [12], revolutionized the teaching of negotiation in many different fields [23], and impacted how practitioners think about negotiation [32]. The other two methods build upon principled negotiation, providing additional guidelines when people insist on uncooperative behaviors [33], or make unacceptable demands or requests [34]. Collectively, the principles and steps of such methods have impacted a wide variety of domains: from health care (e.g., pediatric operating rooms [30]), to personal improvement (e.g., recommendations for raising interpersonal assertiveness [1]), to military (e.g., practical guide for negotiating in the military [7]), and others.

Principled negotiation: Principled negotiation has four principles [9]. First, **separate the people from the problem**: account for differences in thinking, emerging feelings, and misinterpretations during negotiation. Second, **focus on interests, not positions**. Focusing on positions — like predefined deadlines — can create an impasse. The other side's desires and concerns — their interests — helps to uncover other positions that satisfy them. Third, **invent options for mutual gains**. The options should explore differences in parties' interests. A convincing rationale and the consequence for each option also paves the way towards an agreement. Fourth, **insist on using objective criteria**: if interests continue to conflict, objective criteria support reaching an agreement based on principles instead of pressure.

Breakthrough strategy: Ury [33] proposed the breakthrough strategy for situations where people were especially uncooperative. The first step is to **go to the balcony** to suspend one's natural reactions (like giving in) when the other side refuses to reach an agreement. Next, one needs to **step to their side** by hearing them instead of arguing in the face of their disagreement and by expressing one's viewpoints without provocations. The third step is to **reframe**: redirect their attention from the positions to interests, creative options, and fair standards. The fourth step is to **build**

them a golden bridge to walk out from their position to the agreement one wants: ask for constructive criticism of one's proposals or offer them choices to select from. If they keep rejecting, the fifth step is **to use power to educate**: let the other side know the consequences of no agreement.

Positive no: Sometimes, one must say no to a demand or request because they are unacceptable. Poor reactions to such situations are to: (i) accommodate, saying yes when one feels like saying no; (ii) attack, saying no poorly and damaging the relationship; or (iii) avoid, saying nothing at all. Ury [34] proposed the positive no method as an alternative. Its first stage is to *prepare*, starting with to **uncover your Yes**: the reasons why one wants to say No. Next, one needs to **empower your No**, by devising a Plan B: a course of action one will take independently from the other side if they refuse to reach an agreement. The last step to **respect your way to Yes**, by trying to understand the other side utterly. The second stage is to *deliver* one's No. One needs to **express your Yes**, clarifying one's motives for saying No. The next step is to **assert your No**. The final step is to **propose a Yes**, such as a third option to reconcile interests. The third stage is to *follow through*. If the other side rejects the No, the next step is to **stay true to your Yes**: acknowledging their point of view without making concessions. Next, one needs **to underscore your No**. Let reality be their teacher about what will happen if they do not respect one's interests. If necessary, deploy one's Plan B respectfully. Lastly, one needs to **negotiate to Yes**, looking for their interests that might still be unmet by one's proposals.

Related work: McConnell [20] argued in favor of using principled-negotiation [9] in the discussions of estimates, targets, commitments, and plans — also, to defend unpopular schedules [19]. Although the author provided experienced advice in the form of tips, he does not present a structured method completely adapted for the software estimation context. Also, the author did not empirically evaluate the tips. Moreover, other supporting negotiation methods emerged in the last years, such as the breakthrough strategy [33] and the positive no-method [34]. In another study, Ochoa, Pino, and Poblete [24] proposed a method for estimating software cost and duration, claiming to include negotiation as part of it. However, they only focused on internal team consensus, with no guidance about how to deal with external pressures that can lead the team to change their estimates. There is also no discussion about the negotiation theoretical foundations of the technique. To address the gaps from previous research, we proposed the creation and empirical evaluation of an artifact [8] based on the principled negotiation [9], the breakthrough strategy [33], and the positive no method [34]. Next, we describe our artifact and our preliminary results.

3 DEFENSE LENSES FOR THE ESTIMATION CONTEXT

To satisfy the objective of empowering software practitioners in the defense of their software estimates and the negotiation of commitments, we adopted the Design Science Research (DSR) approach [36], as we describe in [8]. We designed an artifact: a set of defense lenses covering the principles and steps from all the negotiation methods that form our foundations (Section 2). We described the whole set of lenses and their supporting examples in a booklet,

which we are now improving as part of our design cycle in DSR. In this section, we present only one lens from our set — presented in Figure 1 — to illustrate the artifact.
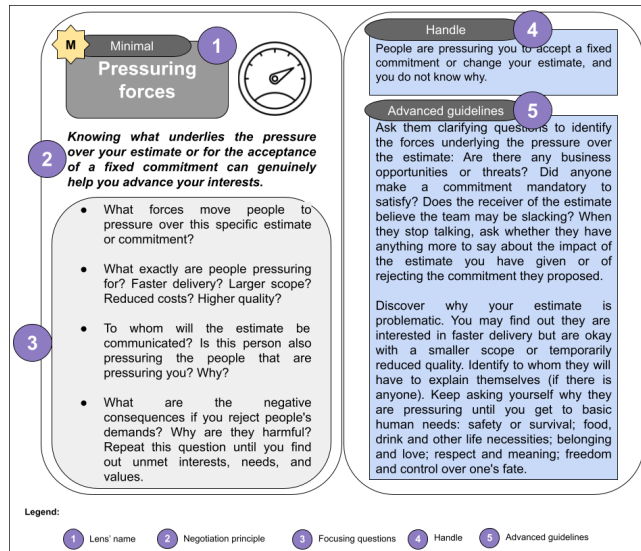

Figure 1: Picture of one of the lenses. A larger and richer version is available athttps://figshare.com/s/01acd2f892f7a56bf18e.

The lenses' format was inspired by the design lenses proposed to guide the design of gamified systems [6]. 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 the lens focus on. Likewise, the defense lenses also have a name (Figure 1.1), a negotiation principle on which we based them (Figure 1.2), and a set of focusing questions (Figure 1.3) on the front side. The focusing questions aim to provoke a mindset shift in estimators: from changing estimates to defending them if no legitimate reasons justify a change, and negotiating commitments that others are trying to impose when they are unrealistic, instead of accepting them promptly. On the backside, we enriched the lens with a handle (Figure 1.4), describing the situations to apply that lens. It helps identify the specific lens we need when using them in isolation. We also added advanced guidelines (Figure 1.5) to support less experienced practitioners or help in more complex situations.

The lens that we exemplified in Figure 1 is entitled *Pressuring Forces*. It is based on the negotiation principle of **focus on interests, not positions** [9] and of listening attentively to the other side to understand their perspective fully —- as part of the **respect your way to Yes** step [34]. The first focusing question aims to investigate the pressuring side's needs and interests, while the second focuses on understanding precisely the what they are pressuring for. The third focusing question regards a particular pressuring force: when someone more powerful is pressing the other side. Identifying this can help the estimator to **step to their side** [33], by opening eyes to ways to relieve their pressure. 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, relating

it to the **respect your way to Yes** step [34]. The advanced guidelines further explore what and how the estimator can investigate the other side's interests.

Except for the Wildcard, which is a particular lens, we organized all the others in two packs: (i) the Minimal Defense Pack and (ii) the Extended Pack. The first encompasses five lenses with core guidance for dealing with specific pressure episodes. The latter provides additional guidance for the situations when the estimator already tried to deflect from a pressure episode using the the Minimal Defense Pack. However, people keep pushing for unjustified changes in the estimates or the acceptance of an unattainable commitment.

Our artifact is also composed of examples that can help even further estimators. For the Pressuring forces lens, we provided a table with a set of detailed examples of clarifying questions that can be useful, including justifications and additional follow-up questions. For instance, a clarifying question is "Does it look like our team can be more productive than this estimate suggests?". The justification is "People may believe the team is unnecessarily padding the estimates, or that lower estimates produce healthy pressure, making the team more productive". The follow-up advice is "Break the tasks to clarify the work dimension".

4 PRELIMINARY RESULTS

In this section, we present the preliminary results regarding the evaluation of our artifact. We carried out a focus group [15], focused on understating about (i) the **perceived usefulness** and (ii) **improvement opportunities** for the design lenses. 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. 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).

In the first focus group meeting, we described the study and the artifact to participants, also providing the booklet for study. In the second meeting, we started by resolving questions from participants and collecting their improvements suggestions. Next, participants engaged in a role-playing activity guided by one scenario similar to the one described in [22, p. 5][1]. One participant played the receptor role (P5 in Table 1), and the others acted as estimators. The estimators worked together to estimate a software project entitled SeminarWeb [14], which we chose after searching for detailed specifications used in previous studies about software effort estimation. Next, all participants got together to define a commitment about the software project, and the receptor pressured estimators to accept a commitment that their estimate did not support, adhering to the scenario description. For this scenario, we expected participants to use the lenses from the Minimal Pack, specially the *Assert your Estimate*, *Pressuring Forces*, and *Candidate Commitments* lenses. After the participants decided they reached the end of the commitment establishment phase, they answered a

---

[1]Supplementary material with the focus group script (including the scenario) is available at https://figshare.com/s/01acd2f892f7a56bf18e.

Table 1: Participants' profile and lenses' uses. The median value from participants is in italics. Values above median are underlined. U = Lens the participant used. Nu = Lens the participant did not understand. Na = Lens the participant decided not to apply. OpE = openness to experience. Con = conscientiousness. Ext = extroversion. Agr = agreeableness. Neu = neuroticism. AsL = assertiveness level. PId = Participant's Id. Y = Years working in the software industry with software engineering activities. Wild = Wildcard. Ladd = Laddering whys. Pres = Pressuring forces. Choo = Choose your battles. Asser = Assert your estimates. Cand = Candidate commitments. Keep = Keep strategy. Pers = Perspective taking. Real = Reality test. Gold = Golden Bridge.

OpE	Con	Ext	Agr	Neu	AsL	PId	Y	Wild	Ladd	Pres	Choo	Asser	Cand	Keep	Pers	Real	Gold
15	*17*	11	*16*	15	43	**P1**	23	Nu	U	U	U	U	U	Na			Nu
14	14	*15*	14	12	*51*	**P2**	1	Nu		Nu		U	U				
6	11	8	14	17	42	**P3**	1					U	U	U			
20	19	17	20	*13*	69	**P4**	12	Nu	U	Na	Na	U	U	Na		Na	
17	18	19	18	11	57	**P5***	23	*P5 acted as the receptor of estimates.									

questionnaire about the lenses and we had a debriefing session for more improvement suggestions.

We had five participants from the software industry, represented by a participants' ID in Table 1. We also collected data regarding participants' **personality traits** (Big-Five Inventory, with 20-items [35]) and **assertiveness** levels (Rathus Assertiveness Scale [27]), because these can be intervening variables in our study. We present participants' profiles and their lenses' uses in Table 1.

Table 1 shows that participants decided to use all lenses from the Minimal Pack, specially 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 P1 chose to use the *Pressuring Forces* lens, and P4 thought it would be better not to apply it at all (Na in Table 1). Also, P2 did not understand it (Nu in Table 1), revealing a need to improve this lens. Also, as we expected, participants decided not use the lenses from the Extended Pack. P4 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.

A few participants indicated they did not understand the *Wildcard*. P1 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, in our booklet, it was the first card to be presented 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. Nevertheless, 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 are still needed.

Another interesting result comes from P3—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, specially more experienced ones. He believes that example sentences

that the estimator can pick from the lens and use directly could help overcome this. Interestingly, P4—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.

5 FUTURE PLANS

In this paper we discuss the relevance of tackling the problem that software practitioners deliberately change their estimates, yielding to pressure over their estimates during the establishment of commitments instead of defending them. We proposed a set of defense lenses: a supporting artifact to empower estimators to deal with this problem. The preliminary results from a focus group provide evidence that the proposed defense lenses are perceived as useful and usable, even for estimators with an assertive profile. Moreover, this endeavor is innovative because it is a concrete step towards supporting software practitioners in developing the relevant soft skill of negotiation, which can be useful in software engineering activities beyond software estimation.

Our next step is to redesign the lenses to include the identified improvements, and to conduct other focus groups, to look for more improvement opportunities. Then, we plan to develop a digital simulation-based training to aid practitioners in acquiring the negotiation skills through the lenses. Digital simulations are promising in promoting self-efficacy beliefs and transfer of training to work [11]. Next, we plan to carry out a field experiment [31], to collect empirical evidence on the practitioners' perceived usefulness of adapting the negotiation methods in the form of the defense lenses in the field, to reach a conclusion about whether our artifact can actually empower them in the defense of their software estimates.

REFERENCES

[1] Daniel Ames, Alice Lee, and Abbie Wazlawek. 2017. Interpersonal assertiveness: Inside the balancing act. *Social and Personality Psychology Compass* 11, 6 (2017), e12317. https://doi.org/10.1111/spc3.12317

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

[3] Luiz Fernando Capretz and Faheem Ahmed. 2018. A Call to Promote Soft Skills in Software Engineering. *Psychology and Cognitive Sciences - Open Journal* 4, 1 (Aug. 2018), e1–e3. https://doi.org/10.17140/PCSOJ-4-e011

[4] Luiz Fernando Capretz, Faheem Ahmed, and Fabio Q. B. da Silva. 2017. Soft sides of software. *Information and Software Technology* 92 (Dec. 2017), 92–94. https://doi.org/10.1016/j.infsof.2017.07.011 Publisher: Elsevier.

[5] Aldo Dagnino. 2013. Estimating software-intensive projects in the absence of historical data. In *2013 35th International Conference on Software Engineering (ICSE)*. 941–950. https://doi.org/10.1109/ICSE.2013.6606643 ISSN: 1558-1225.

[6] Sebastian Deterding. 2015. The lens of intrinsic skill atoms: A method for gameful design. *Human-Computer Interaction* 30, 3-4 (2015), 294–335. https://doi.org/10.1080/07370024.2014.993471 Place: United Kingdom Publisher: Taylor & Francis.

[7] Stefan Eisen Jr. 2017. *Practical Guide to Negotiating in the Military*. Technical Report AD1117925. https://apps.dtic.mil/sti/citations/AD1117925

[8] Patrícia Gomes Fernandes Matsubara. 2019. Dealing with software estimates distortions from the perspective of negotiation theories. *ACM SIGSOFT Software Engineering Notes* 44, 3 (Nov. 2019), 22. https://doi.org/10.1145/3356773.3356794

[9] Roger Fisher, William Ury, and Bruce Patton. 2011. *Getting to Yes: Negotiating Agreement Without Giving in* (3rd revised ed.). Penguin Books.

[10] Vahid Garousi, Görkem Giray, Eray Tüzün, Cagatay Catal, and Michael Felderer. 2019. Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software* 156 (Oct. 2019), 65–83. https://doi.org/10.1016/j.jss.2019.06.044

[11] Andreas Gegenfurtner, Carla Quesada-Pallarès, and Maximilian Knogler. 2014. Digital simulation-based training: A meta-analysis. *British Journal of Educational Technology* 45, 6 (2014), 1097–1114. https://doi.org/10.1111/bjet.12188

[12] Frederik Reinder Hak and Karin Sanders. 2018. Principled negotiation: an evidence-based perspective. *Evidence-based HRM: a Global Forum for Empirical Scholarship* 6, 1 (Jan. 2018), 66–76. https://doi.org/10.1108/EBHRM-03-2017-0014 Publisher: Emerald Publishing Limited.

[13] Torleif Halkjelsvik and Magne Jørgensen. 2018. Time Prediction Biases. In *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*, Torleif Halkjelsvik and Magne Jørgensen (Eds.). Springer International Publishing, Cham, 55–70. https://doi.org/10.1007/978-3-319-74953-2_5

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

[15] Jyrki Kontio, Johanna Bragge, and Laura Lehtola. 2008. The Focus Group Method as an Empirical Tool in Software Engineering. In *Guide to Advanced Empirical Software Engineering*, Forrest Shull, Janice Singer, and Dag I. K. Sjøberg (Eds.). Springer, London, 93–116. https://doi.org/10.1007/978-1-84800-044-5_4

[16] Miikka Kuutila, Mika V Mantyla, Umar Farooq, and Maelick Claes. 2021. What Do We Know About Time Pressure in Software Development? *IEEE Software* 38, 5 (Sept. 2021), 32–38. https://doi.org/10.1109/MS.2020.3020784

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

[18] Patricia Matsubara, Igor Steinmacher, Bruno Gadelha, and Tayana Conte. 2021. Buying time in software development: how estimates become commitments?. In *Proceedings of the 14th International Conference on Cooperative and Human Aspects of Software Engineering*. Virtual, 61–70.

[19] S. McConnell. 1996. How to defend an unpopular schedule [software development projects]. *IEEE Software* 13, 3 (May 1996), 120–119. https://doi.org/10.1109/52.493033

[20] Steve McConnell. 2006. Politics, Negotiation, and Problem Solving. In *Software Estimation: Demystifying the Black Art*. Microsoft Press, Redmond, 259–270. Google-Books-ID: iXghAQAAIAAJ.

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

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

[23] Carrie Menkel-Meadow. 2006. Why Hasn't the World Gotten to Yes? An Appreciation and Some Reflections. *Negotiation Journal* 22, 4 (2006), 485–503. https://doi.org/10.1111/j.1571-9979.2006.00119.x

[24] Sergio F. Ochoa, José A. Pino, and Fábio Poblete. 2009. Estimating Software Projects Based On Negotiation. *Journal of Universal Computer Science* 15 (2009), 1812–1832.

[25] John R. Ogilvie and Deborah L. Kidder. 2008. What about negotiator styles? *International Journal of Conflict Management* 19, 2 (April 2008), 132–147. https://doi.org/10.1108/10444060810856076

[26] Robert Ramač, Vladimir Mandić, Nebojša Taušan, Nicolli Rios, Manoel G. de Mendonca Neto, Carolyn Seaman, and Rodrigo Oliveira Spínola. 2020. Common Causes and Effects of Technical Debt in Serbian IT: InsighTD Survey Replication. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 354–361. https://doi.org/10.1109/SEAA51224.2020.00065

[27] Spencer A. Rathus. 1973. A 30-item schedule for assessing assertive behavior. *Behavior Therapy* 4, 3 (May 1973), 398–406. https://doi.org/10.1016/S0005-7894(73)80120-0

[28] Nicolli Rios, Rodrigo Oliveira Spínola, Manoel Mendonça, and Carolyn Seaman. 2020. The practitioners' point of view on the concept of technical debt and its causes and consequences: a design for a global family of industrial surveys and its first results from Brazil. *Empirical Software Engineering* 25, 5 (Sept. 2020), 3216–3287. https://doi.org/10.1007/s10664-020-09832-9

[29] Andrea Kupfer Schneider, Noam Ebner, David Matz, and John Lande. 2017. The Definition of Negotiation: A Play in Three Acts. *Missouri Journal of Dispute Resolution* 1 (2017). https://doi.org/10.2139/ssrn.2902743

[30] Jina L. Sinskey, Joyce M. Chang, Gail S. Shibata, Andrew J. Infosino, and Kathryn Rouine-Rapp. 2019. Applying Conflict Management Strategies to the Pediatric Operating Room. *Anesthesia and Analgesia* 129, 4 (Oct. 2019), 1109–1117. https://doi.org/10.1213/ANE.0000000000003991

[31] Klaas-Jan Stol and Brian Fitzgerald. 2020. Guidelines for Conducting Software Engineering Research. In *Contemporary Empirical Methods in Software Engineering*, Michael Felderer and Guilherme Horta Travassos (Eds.). Springer International Publishing, Cham, 27–62. https://doi.org/10.1007/978-3-030-32489-6_2

[32] Chia-Jung Tsay and Max H. Bazerman. 2009. A Decision-Making Perspective to Negotiation: A Review of the Past and a Look to the Future. *Negotiation Journal* 25, 4 (2009), 467–480. https://doi.org/10.1111/j.1571-9979.2009.00239.x

[33] William Ury. 2007. *Getting Past No: Negotiating in Difficult Situations* (revised ed.). Bantam.

[34] William Ury. 2012. *The Power of A Positive No*. Hodder & Stoughton.

[35] Valdiney Veloso Gouveia, Rafaella de Carvalho Rodrigues Araújo, Isabel Cristina Vasconcelos de Oliveira, Marina Pereira Gonçalves, Taciano Milfont, Gabriel Lins de Holanda Coelho, Walberto Santos, Emerson Diógenes De Medeiros, Ana Karla Silva Soares, Renan Pereira Monteiro, Josemberg Moura de Andrade, Thiago Medeiros Cavalcanti, Bruna Da Silva Nascimento, and Rildésia Gouveia. 2021. A Short Version of the Big Five Inventory (BFI-20): Evidence on Construct Validity. *Revista Interamericana de Psicología/Interamerican Journal of Psychology* 55, 1 (April 2021), e1312. https://doi.org/10.30849/ripijp.v55i1.1312

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