# A Systematic Mapping of Software Requirements Negotiation Techniques

Lucas Tito, Alexandre Estebanez, Andréa Magalhães Magdaleno,
Daniel de Oliveira, Marcos Kalinowski
*Computing Institute, Fluminense Federal University (UFF), Av. Gal. Milton Tavares de Souza, Niteroi , Brazil*
{*lucastito@id.uff.br, aestebanez@id.uff.br, andrea@ic.uff.br, danielcmo@ic.uff.br, kalinowski@ic.uff.br*

Keywords: Software Requirements Negotiation Techniques, Requirements Engineering

Abstract: [Context] Eliciting requirements is a commonly discussed task. However, after they are ready, it is essentially important for a software project that these requirements are sufficient for stakeholders to reach their goals. Therefore, techniques to negotiate schedule, price, quality, and scope among stakeholders are important. [Goal] This paper aims at identifying and presenting characteristics of techniques that have been proposed and/or used to negotiate software requirements. [Method] A mapping study was planned and conducted to identify techniques and to capture their characteristics. Those characteristics include description, environment (*e.g.* academic, industrial), the types of research being published, and the types of primary studies. The main findings of the papers, and the advantages and disadvantages reported for theses techniques were also summarized. [Results] We mapped the characteristics of 10 different requirements negotiation techniques identified in 33 papers which met our inclusion criteria. We found that most of the identified techniques can be seen as variations of the seminal WinWin requirements negotiation technique proposed in 1994. [Conclusions] The conducted mapping study provides an interesting overview of the area and may also be useful to ground future research on this topic.

## 1 INTRODUCTION

Requirement engineering involves a wide variety of stakeholders (*e.g.*, users, customers, developers, project managers) and difficulties associated with the communication between them (Méndez Fernández et al., 2016). Accordingly to (Schopenhauer, 1831), men are moved by the idea of selfishness. Considering this fact and the different goals of the stakeholders, it is easy to understand why conflicts are common. Another important aspect is that when bonds are made and relationships of trust are established, people (in our context software project stakeholders) will try harder to find consensus during negotiations on their different dimensions of interest (Le Bon, 1895).

The different dimensions of interest present in software requirements negotiations are *Level of Service, Capabilities, Schedule, Cost,* and *Productivity* (Grünbacher and Briggs, 2001). Depending on the situation, each dimension can be variable , but the change in one of them will always affect the others.

Hence, it is difficult to find the ideal trade-off between the dimensions and software requirements negotiation is fundamental to the project's success (Boehm et al., 1995). There are many different techniques to conduct software requirements negotiation, but currently there is no clear overview of the available ones. Therefore, we conducted a mapping study on those techniques to provide such overview.

To allow for a precise understanding what we mean with a software negotiation technique in the context of this paper, we created the following definition: a series of steps, method and rules that are applied by two or more people trying to reach a consensus and establish agreements that minimize the differences in their points of view regarding the software requirements. It is noteworthy, we only understand techniques as requirements negotiation techniques if they are designed to be conducted by two or more people. This excludes other techniques, such as solo requirements prioritization techniques.

The goal of this paper is to identify and to provide an overview of requirements negotiation techniques, published in academic literature, and their characteristics. The main findings of the papers and the advantages and disadvantages reported for theses techniques are also presented and discussed.

The remainder of this paper is organized as follows. In Section 2, we describe our research questions, systematic mapping planning, execution and the identified papers. In Section 3, we discuss the results and proceed to answer our research questions. Finally, Section 4 presents the concluding remarks.

# 2 SYSTEMATIC MAPPING

This section, presents the planning, execution and the identified papers of the systematic mapping study. In order to conduct the mapping on software requirements negotiation techniques in a wide and unbiased way, we used the definition presented in the introduction and built our protocol according to the guidelines defined by (Kitchenham and Charters, 2007).

## 2.1 PLANNING

Our systematic mapping study addresses one main research question named RQ1: ”Which techniques have been proposed and(or) used to negotiate software requirements?”

Software requirement negotiation techniques may be defined, applied or evaluated in many different contexts. To better understand the context where these different techniques are mentioned, we propose to also analyze the following secondary research questions: (i) SQ1. ”Where the requirement negotiation techniques were defined, applied or evaluated, in industry or in academy?”; (ii) SQ2. ”What are the types of research about requirement negotiation techniques that have been published? (*e.g.*, *Evaluation research, Solution proposal, Philosophical paper, Opinion paper, Experience paper*)?”; (iii) SQ3. ”What are the types of primary studies that have been used to evaluate requirement negotiation techniques? (*e.g.*, experiment, case study, survey)?”; (iv) SQ4. ”What are the advantages and disadvantages of each technique according to the literature?”; and (v) SQ5. ”What is the knowledge (main findings) found in the literature about each technique?”.

The types of research mentioned in SQ2 were obtained from (Wieringa et al., 2005) and can be described as follows: (i) **Evaluation research:** implemented in practice, the evaluation of the implementation; demands more than one demonstration case study.; (ii) **Solution proposal:** Solution for a problem is proposed, benefits/application is demonstrated by example, experiments, or student labs; (iii) **Philosophical paper:** New way of thinking, structuring a field in form of a taxonomy or a framework, secondary studies like SLR or SMS; (iv) **Opinion paper:** Personal opinion, not grounded in related work and research methodology; and (v) **Experience paper:** Personal experience, how things are done in practice.

Our search strategy to identify relevant papers describing requirements negotiation techniques included conducting a search string-based query and complementing the results by applying backward snowballing (searching for new papers in the references). The elaboration of the search string followed the PICO strategy, as suggested by (Kitchenham and Charters, 2007).

The *Population* is defined as software projects (keyword *software*); the *Intervention* is what we are observing within the population - requirements negotiation techniques (keywords *scope*, *requirements* and *negotiation*). As common for mapping studies, there is no *Comparison* and no definition for the *Outcome* of the primary studies. The resulting search string follows: **((”*software*”) AND (”*scope*” OR ”*requirements*”) AND (”*negotiation*”))**.

For filtering the papers we defined the following inclusion and exclusion criteria (Table 1). Those criteria should be applied by two independent researchers. In case of differences in their classification a third researcher should be involved to also evaluate to reach final consensus.

To check the efficiency of our string-based query, we manually identified 3 relevant papers that attend the inclusion and exclusion criteria and are used as control papers: (i) *Software requirements negotiation using the software quality function deployment* (Ramires et al., 2005); (ii) *Integration of scrum with Win-Win requirements negotiation model* (Khan et al., 2014); (iii) *Requirements Negotiation Using Multi-Criteria Preference Analysis* (In and Olson, 2004).

Table 1: Inclusion and exclusion criteria

| **Criterion** | **Description** |
|---|---|
| Inclusion (IC1) | Papers that focus on requirements negotiation techniques and describe, propose, apply or evaluate one of such techniques. |
| Exclusion (EC1) | Papers in which the technique is only mentioned or that do not describe the technique steps in a understandable way. |
| Exclusion (EC2) | Papers that are not indexed in digital libraries and available online. |
| Exclusion (EC3) | Papers in any language other than english. |
| Exclusion (EC4) | Papers that describe requirement negotiation techniques made for non face to face contexts. |

The papers found in the query result should be analyzed by title and abstract, taking into account the proposed inclusion and exclusion criteria. This analysis should naturally result into two groups: the approved papers and the rejected ones. The first group is the starting point for the next filtering phase. In this step, the researchers read and evaluate each paper using the aforementioned criteria to decide whether the paper is aligned with the research goal or not. The resulting papers from this step are considered in a backward snowballing process, which significantly contributes to finding relevant research in the literature (Badampudi et al., 2015). With the articles that had been found by applying the snowballing, we repeated the same filtering processes that was done previously. The snowballing and filtering are iterative, *i.e.* they stop when there are no more results to be added.

We extract the following data from the included papers: (i) Reference; (ii) Technique name; (iii) Technique description summary; (iv) Environment (academic or industrial); (vi) Type of research (*Evaluation research, Solution proposal, Philosophical paper, Opinion paper, Experience paper)*; (vii) Type of primary study (Experiment, Case study or Survey), if applicable; (viii) Main findings of the paper (knowledge found); (ix) Advantages; and (x) Disadvantages.

## 2.2 EXECUTION

For conducting our search strategy we selected Scopus, which claims to be the largest database of abstracts and citations (Kitchenham and Charters, 2007). Applying the filtering to the papers retrieved by Scopus should allow us to obtain a large and unbiased set of papers that could be used as a representative seed set to start backward snowballing (Badampudi et al., 2015). We used the search string to conduct the query in Scopus, limiting the results to the areas of *computer science* and *engineering*.

The search in the library was performed in November 24th 2015 and retrieved a set of 408 papers. These papers were read and filtered by the first two authors following the aforementioned protocol, consulting the last author in case of classification differences. The set obtained after the first round of filtering (by title and abstract) contained 39 papers, and the one obtained after the second round of filtering contains 18 papers.

Control papers (1) and (3) were successfully retrieved. We observed that the control paper (2) was not retrieved due to the area restriction (computer science and engineering), which intended to avoid papers from areas like economics . The reason was that the paper (2) was classified as *multiarea*. This reinforced our confidence of having a good search string for retrieving this kind of paper. Despite this indexing issue, control paper (2) was considered and its data was extracted and included in the set of 18 papers.

During the first iteration of the backward snowballing process using these 18 papers as seed set, a total of 383 papers were found. After the filtering by title, the set was reduced to 56 non-duplicated papers. The abstract filtering reduced this amount to 26. After analyzing the content of each paper we had 15 additional papers that passed our criteria. During the second iteration, within the 332 references of these 15 additional papers, 7 non-duplicated papers that had not been analyzed before remained after the filter by title. Out of these 7, 3 were excluded by reading the abstract and the other 4 had promising titles but had to be excluded using criterion *EC2*. Thus, our final set remained with 33 papers that met our criteria and that had their data extracted.

The identified papers, their publication years and the references among them are shown in the citation graph available at https://goo.gl/ZjG2bv. It depicts the 18 identified papers retrieved from Scopus as the seed set in green. The 15 papers included after the backward snowballing are shown in blue. Based on the content of these papers we answer our research questions in Section 3.

# 3 RESULTS AND DISCUSSION

The extracted data and the knowledge acquired after reading the papers subsidize the answers to our research questions.

## 3.1 REQUIREMENT NEGOTIATION TECHNIQUES (RQ1)

A total of 10 requirement negotiation techniques have been identified in the 33 included papers. The techniques are shown in Table 2 together with its reference. The techniques are described hereafter ordered by their chronological proposal.

(I) **WinWin:** It is based on the theory W (Boehm et al., 1994). Four fundamental concepts are applied, which are: *Win Conditions*, *Issues*, *Options* and *Agreements*. *Win Conditions* capture the stakeholder goals and concerns about the new system. An *Agreement* is created when a *Win Condition* does not conflict with others. On the other hand, when there are conflicts, *Issues* are recorded and the stakeholders could suggest alternative solutions about these *Issues*, which are represented by *Options* and could be used by an *Agreement* to resolve *Issues*. The WinWin steps are: (i) Stakeholders identification; (ii) Stakeholders' *Win Conditions* detection; and (iii) Reconciliation negotiation of *Win Conditions*. Once this technique is applied by using the spiral model, a new requirement can be introduced and it usually unbalances the WinWin status, thus the technique needs to undergo a modification to include some steps such as: validating and verifying what has been negotiated, analyzing the generated conflicts and the risks for the next iteration.

(II) **Easy WinWin:** It has the same steps as WinWin technique. Its main differential is stakeholders collaboration during the negotiation process. The stakeholders prioritization about the business importance and the ease of realization are needed.There are 4 prioritization categories: (i) Low hanging fruits: Win Conditions that are perceived as important and where the expected difficulties for realization are moderate; (ii) Important with hurdles: Important Win Conditions that are difficult to implement; (iii) Maybe later: Unimportant Win Conditions that are attractive because of their burden-free implementation are labeled; and (iv) Forget them: Unimportant Win Conditions that are difficult to achieve are tagged. When it comes to negotiation environment, stakeholders often forget a Win Condition.

(III) **CBSP (Component-Bus-System-Property):** it is a lightweight approach intended to provide a systematic way of reconciling requirements and architectures. It aims at transforming requirements descriptions into architectural descriptions to further understand and reduce technical conflicts. These conflicts arise from the interpretation of requirements among software architects. CBSP's steps are the following: (i) Selection of next level requirements; (ii) Architectural classification of requirements; (iii) Identification and resolution of classification mismatches; (iv) Architectural refinement of requirements; and (v) Derivation of architectural style and architecture.

(IV) **MPARN (Multi-Criteria Preference Analysis Requirements Negotiation):** it is also based on the WinWin model. However, it is more objective and systematic than WinWin. MPARN applies the systematic analysis of the preferences by using weights and points. For the purpose of defining the criteria it is necessary to implement the following steps: (i) Eliciting *Win Conditions*; (ii) Identifying issues and conflicts; (iii) Exploring conflict-resolution options; (iv) Exploring objective criteria; (v) Assessing options based on the criteria; (vi) Assessing relative weights for criteria by each stakeholder; (vii) Ranking options; and (viii) Post-analysis for agreements.

(V) ***Quantitative WinWin*****:** It is very similar to the WinWin technique, however the former uses Analytic Hierarchy Process (AHP) to classify the requirements and the stakeholders and measuring their importance. In the end, a sequence of iterations of analysis is performed to determine the optimal result.

(VI) ***QA EasyWinWin*****:** It is a gathering of Easy WinWin with quality assurance practices, so it has all of Easy WinWin's steps and still involves applying some quality assurance techniques before, during and after the negotiation.

(VII) ***Collaborative QFD (Quality Function Deployment)*****:** It involves a matrix of correlation values between requirements and specifications. This matrix is used in the following way: (i) users' requirements are elicited to relevant stakeholders and placed in the left-hand side (rows); (ii) with the help of the stakeholders, the requirements are converted to technical specifications and placed at the upper side (columns); (iii) the stakeholders are then invited to complete the matrix with their perceived correlations; (iv) a list of requirements priorities is defined; and (v) a list of technical specifications priorities is defined. This one can be considered a requirement negotiation technique when stakeholders participate in meetings collaboratively to define values for the matrix.

(VIII) **Win CBAM:** It's goal is to reconcile with focus on cost-benefit of the options. It establishes the following steps: (i) Eliciting win conditions; Identifying conflict Issues; (ii) Exploring options / architecture strategies; (iii) Assessing quality assurance benefits; (iv) Quantifying the architecture strategies benefits; (v) Quantifying the architecture strategies cost / schedule implications; (vi) Calculating desirability; and (vii) Reaching agreements.

(IX) ***Requirement Negotiation Spiral Model*****:** It is similar to WinWin, but some steps have different semantics and there are some assumptions for them to be applied. The steps are: (i) Identify Conflicts; (ii) Develop Alternatives Solutions; (iii) Elaborate Solutions; (iv) Judgment and Trade-off; (v) Evaluate and Analyze Agreement. In order to use the technique, 3 external elements are needed: scenario, criterion, and resolution strategy. The scenario consists of a stakeholders mapping, the criterion would be the way how the requirements are prioritized and the strategy is one archetype of conflict resolution.

(X) ***WinWin with Scrum*****:** It is based on WinWin Technique and therefore has the same steps. However, it defines roles, the order of meetings, and the use of artifacts in the same way as Scrum. The main difference is that there must be a negotiation meeting for each *Sprint*, whereas in WinWin, the meetings do not have a defined schedule and well defined roles. The *Product Backlog* and *Sprint Backlog* are adapted to contain requirements. The *PMO* defines the requirements together with the clients, the *Master Negotiator* ensures the proper functioning of the meetings and *Negotiation Team* raise points for discussion.

Table 2: Identified techniques and references

| Technique | Reference |
|---|---|
| WinWin | (Boehm et al., 1994); (Boehm et al., 1995); (Lee, 1996); (Boehm et al., 1997); (Boehm and Egyed, 1998); (Boehm and Egyed, ); (Boehm et al., 1998a); (Boehm et al., 1998b); (Egyed and Boehm, 1999); (Boehm et al., 2001); (In et al., 2001); (Ruhe, 2002); (Raja et al., 2007); (Boehm and Kitapci, 2006); (Grünbacher et al., 2007); (Khan et al., 2014) |
| Easy WinWin | (Egyed and Boehm, 1997); (Grünbacher, 2000); (Grünbacher and Boehm, 2001); (Grünbacher and Briggs, 2001); (Stallinger and Grünbacher, 2001); (Briggs and Grünbacher, ); (Grünbacher and Braunsberger, 2003); (Grünbacher and Seyff, 2005); |
| CBSP | (Grünbacher et al., 2001); |
| MPARN | (In et al., 2002); (In and Olson, 2004) |
| Quantitative WinWin | (Ruhe et al., 2002); (Ruhe et al., 2003); |
| QA EasyWin-Win | (Grünbacher et al., 2004b); |
| Collaborative QFD | (Ramires et al., 2005); |
| Win CBAM | (Kazman et al., 2005) |
| Requirement Negotiation Spiral Model | (Ahmad, 2008); |
| WinWin with Scrum | (Khan et al., 2014); |

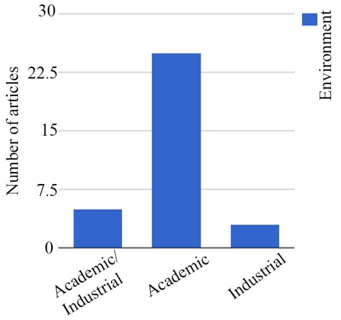

Figure 1: Environment of the identified research

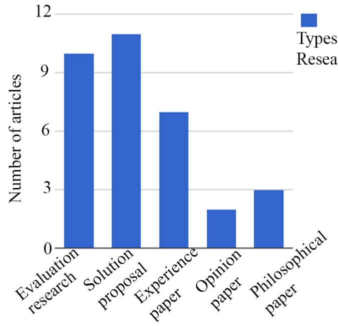

Figure 2: Types of research

## 3.2 ENVIRONMENTS, TYPES OF RESEARCH AND TYPES OF PRIMARY STUDIES

Concerning SQ1 (environment), most techniques have been applied only in academic environments (Figure 1) in order to improve the techniques and/or software engineering education. Thus, apparently little is known about the adherence and acceptance of these techniques by industry.

Regarding SQ2 (research type), it is possible to observe (Figure 2) that solution proposals and evaluation research are predominant, closely followed by experience papers.

Out of the papers containing primary studies (SQ3), most were case studies conducted in the academic context (e.g., tasks to be completed using the negotiation techniques), although some were also controlled experiments conducted by professors with the participation of groups of students (Figure 3).

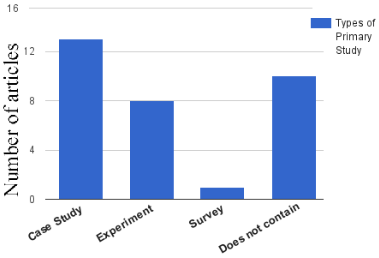

Figure 3: Types of primary studies

## 3.3 ADVANTAGES, DISADVANTAGES AND MAIN FINDINGS

The purpose of this subsection is to answer SQ4 and SQ5, providing information on advantages, disadvantages, and findings regarding the techniques.

### 3.3.1 ADVANTAGES

The WinWin, when applied by using the Spiral Model, and the WinWin with Scrum are reported to be able to handle changes and accept new requirements in any step of the negotiation. (Boehm et al., 2001)(Khan et al., 2014). One cited advantage concerning WinWin with Scrum and Easy WinWin is a shorter time to reach the negotiation goals (Khan et al., 2014)(Briggs and Grünbacher, ). The Easy WinWin and the WinWin techniques are reported to have a very effective approach for communication and collaboration between two or more stakeholders. (Grünbacher and Briggs, 2001)(Boehm and Egyed, ). According to Grünbacher (2000), the Easy WinWin technique also avoids stakeholders to be judged by their opinions.

The Win CBAM, the CBSP and the QA Easy WinWin techniques include steps for assessing the agreement between two or more stakeholders (Kazman et al., 2005)(Grünbacher et al., 2001)(Grünbacher et al., 2004b). These last two techniques also are mentioned to promote the comprehension between distinct viewpoints (technique vs functional) (Grünbacher et al., 2001)(Grünbacher et al., 2004b). On the other hand, QA Easy WinWin and WinWin present domain flexibility (Grünbacher et al., 2004b)(Boehm et al., 1997). The Quantitative WinWin and MPARN techniques, differently from all others, focus on adapting the WinWin technique to obtain a decision-making in a more quantitative and objective way (Ruhe et al., 2002)(In and Olson, 2004).

#### 3.3.2 DISADVANTAGES

Most of the analyzed papers do not directly mention disadvantages. Indeed, these papers, in some way, seem to recommend the use of the presented techniques. This may be due to publication bias. Consequently, some of the following disadvantages are the result of the interpretation by the authors.

In the WinWin technique, at the stage of negotiation, a stakeholder might not feel fully satisfied with the agreements, because they may have accepted more than they would like to.

Since the WinWin technique is subjective, it might be necessary to organize more negotiation meetings to analyse a set of requirements when compared to using Quantitative WinWin (Ruhe, 2002).

The Quantitative WinWin technique and MPARN, on the other hand, are more objective and are highly dependent on tools, whereas it is possible loosing focus during the requirements comparison assignments, in cases where there are many requirements to be analyzed. In this regard, the stakeholders need to be guided to not forget to perform comparisons. The MPARN helps to make local decisions, however, it is not completely effective for global decisions (Ruhe et al., 2002)(In and Olson, 2004). It is noteworthy that we did not identify studies evaluating the Quantitative WinWin and MPARN in practical industrial contexts. Therefore, their adoption might still be associated to unknown risks. In Collaborative QFD, the larger the matrix and more stakeholders, the shorter the time for decision making and the longer the total time of the negotiation process (Ramires et al., 2005).

#### 3.3.3 MAIN FINDINGS

The main findings can be summarized as follows: (i) Users and clients were more active in defining win conditions, while developers and analysts were more active in resolving issues and trying to find options (In et al., 2001); (ii) There are several ways to apply MPARN due to a lot of kinds of mathematical formulas in its steps (In et al., 2002); (iii) Easy WinWin helps in the software documentation and it is interesting to associate it with a software inspection technique (Khan et al., 2014); (iv) Stakeholders tend to accept satisfying Win Conditions. Once they achieved them, they do not engage to obtain an optimal solution for themselves (In et al., 2001); (v) It is important to develop prototypes during the negotiation and the software development process, so that users and customers do not change their minds at the end of negotiations (Boehm et al., 1998a); (vi) A lot of Win Conditions can be achieved, when brainstorming is used (Grünbacher and Briggs, 2001); and (vii) Face to face negotiation provide a sense of confidence and security to stakeholders (Egyed and Boehm, 1997).

In some papers, WinWin and Spiral WinWin are considered as requirements negotiation models (more abstract then techniques). They are described as an outline that defines the general negotiation steps.

Indeed, most of the identified techniques can be seen as variations of WinWin (e.g., WinWin with Scrum, Quantitative WinWin and MPARN are basically WinWin with mathematical models, EasyWinWin is based on WinWin with a stronger focus on collaboration, QA Easy WinWin extends EasyWinWin with quality assurance practices, Win CBAM is basically WinWin applied to software components).

### 3.4 LIMITATIONS AND THREATS TO VALIDITY

As any study, ours also faces limitations and threats to validity. The first one is a lack of a clear taxonomy for requirements negotiation techniques. For instance, we identified papers in which negotiation was defined as prioritization, involving just one stakeholder. In these cases, we did not consider it as a negotiation technique. In other cases, authors used the word *framework* for different meanings (*e.g.*, system, tool, model, technique). To minimize this threat, before building our mapping protocol, we agreed on a precise definition of what a requirements negotiation technique represents for us (cf. Section 1).

The filtering process is effort intensive and involves human judgement. In many cases the title and abstract could lead us to exclude a paper due to not allowing a precise understanding that the focus of the paper is on requirements negotiation techniques. Although this threat can not be directly handled, the filtering process was conducted independently by two researchers and a third researcher was involved to solve eventual differences.

Forward snowballing (Felizardo et al., 2016) was not applied. This was mainly because the quantity of papers citing the 33 identified papers would be too large to be handled with reasonable effort. For instance, according to Google Scholar, (Boehm et al., 1998a) (Boehm et al., 1995) (Boehm et al., 1994) (Grünbacher and Briggs, 2001) papers alone were cited in almost 1000 papers. Thus, some more recent papers might not have been identified, given that we used only one digital library (even though it claims being the largest).

Hence, as an additional quality assurance mechanism we identified the three authors with most publications among the 33 identified papers (Barry Boehm, Paul Grünbacher, and Alexander Egyed, involved in 22 of the 33 identified papers) and verified within the titles of their online DBLP computer science bibliography if there were additional papers to be included. This additional verification improved our confidence in the reliability of the results. During this process we found only three additional papers: (Grünbacher et al., 2004a), (Kitapci and Boehm, 2007), and (Vogl et al., 2011). The first one was not indexed in Scopus. The other two did not include the word *software* in the title, abstract or keywords. However, the word *software* was needed to add precision to the search string. All three would have been retrieved applying forward snowballing, since they all cite some of the 33 identified papers. The effort of applying forward snowballing to complement the results will be the focus of a future work.

## 4 CONCLUSIONS

This paper describes a systematic mapping about software requirements negotiation techniques. Its contributions are the protocol planning and the mapping results. We identified 33 papers that describe 10 different requirements negotiation techniques. For each of these techniques we extracted and summarized their information. Additionally, we highlighted findings mentioned in the identified papers, and reported advantages and disadvantages of the techniques.

We believe that these pieces of information are useful to both, researchers and practitioners. Researchers might obtain an overview of the research conducted concerning requirement negotiation techniques. The practitioners in turn might use the results to help them to choose a technique. It is important to emphasize that the negotiation of requirements is an inherent activity to the software development process, that happens with or without a technique, although choosing a technique may improve this process.

Also, we identified a lack of evaluations conducted in industry. Therefore, future work could involve conducting empirical studies in industry. Also, a survey with practitioners could allow us to verify whether the industry is aware of the identified techniques (or others) and if it already uses some of them.

## REFERENCES

Ahmad, S. (2008). Negotiation in the requirements elicitation and analysis process. In *Proceedings of the 19th ASWEC*, pages 683–689.

Badampudi, D., Wohlin, C., and Petersen, K. (2015). Experiences from using snowballing and database searches in systematic literature studies. In *Proc. of EASE*, pages 17:1–17:10.

Boehm, B., Bose, P., Horowitz, E., and Lee, M.-J. (1994). Software requirements as negotiated win conditions. In *Proc. of ICRE*, pages 74–83.

Boehm, B., Bose, P., Horowitz, E., and Lee, M. J. (1995). Software requirements negotiation and renegotiation aids: A theory-w based spiral approach. In *Proc. of ICSE*, pages 243–243.

Boehm, B. and Egyed, A. Winwin requirements negotiation processes: A multi-project analysis. In *Proc. of ICSP*.

Boehm, B. and Egyed, A. (1998). Software requirements negotiation: Some lessons learned. In *Proc. of ICSE*, pages 503–506.

Boehm, B., Egyed, A., Kwan, J., and Madachy, R. (1997). Developing multimedia applications with the winwin spiral model. In *Proc. of FSE*, pages 20–39.

Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., and Madachy, R. (1998a). Using the winwin spiral model: A case study. *Computer*, 31(7):33–44.

Boehm, B., Egyed, A., Port, D., Shah, A., Kwan, J., and Madachy, R. (1998b). A stakeholder win–win approach to software engineering education. *Annals of Software Engineering*, 6(1):295–321.

Boehm, B., Grünbacher, P., and Briggs, R. O. (2001). Developing groupware for requirements negotiation: Lessons learned. *IEEE Software*, 18(3):46–55.

Boehm, B. and Kitapci, H. (2006). The winwin approach: using a requirements negotiation tool for rationale capture and use. In *Rationale Management in Software Engineering*, pages 173–190. Springer Berlin Heidelberg.

Briggs, R. O. and Grünbacher, P. Easywinwin: Managing complexity in requirements negotiation with gss. In *Proc. of HICSS*.

Egyed, A. and Boehm, B. (1997). Analysis of system requirements negotiation behavior patterns. In *Proc. of INCOSE*, pages 481–488.

Egyed, A. and Boehm, B. (1999). Comparing software system requirements negotiation patterns. *Systems Engineering*, 2(1):1–14.

Felizardo, K. R., Mendes, E., Kalinowski, M., Souza, E. F., and Vijaykumar, N. L. (2016). Using forward snowballing to update systematic reviews in software engineering. In *Proc. of ESEM*, pages 53:1–53:6.

Grünbacher, P. (2000). Collaborative requirements negotiation with easywinwin. In *Proc. of DEXA*, pages 954–958.

Grünbacher, P. and Boehm, B. (2001). Easywinwin: A groupware-supported methodology for requirements negotiation. *SIGSOFT Softw. Eng. Notes*, 26(5):320–321.

Grünbacher, P. and Braunsberger, P. (2003). Tool support for distributed requirements negotiation. *Cooperative methods and tools for distributed software processes*, pages 56–66.

Grünbacher, P. and Briggs, R. O. (2001). Surfacing tacit knowledge in requirements negotiation: experiences using easywinwin. In *Proc. of HICSS*.

Grünbacher, P., Egyed, A., and Medvidovic, N. (2001). Reconciling software requirements and architectures: The cbsp approach. In *Proc. of RE*, pages 202–211.

Grünbacher, P., Egyed, A., and Medvidovic, N. (2004a). Reconciling software requirements and architectures with intermediate models. *Software & Systems Modeling*, 3(3):235–253.

Grünbacher, P., Halling, M., Biffl, S., BIFFL, S., and Boehm, B. W. (2004b). Integrating collaborative processes and quality assurance techniques: experiences from requirements negotiation. *Journal of Management Inf. Sys.*, 20(4):9–29.

Grünbacher, P. and Seyff, N. (2005). Requirements negotiation. In *Engineering and managing software requirements*, pages 143–162. Springer Berlin Heidelberg.

Grünbacher, P., Seyff, N., Briggs, R. O., In, H. P., Kitapci, H., and Port, D. (2007). Making every student a winner: The winwin approach in software engineering education. *JSS*, 80(8):1191 – 1200.

In, H., Boehm, B., Rodgers, T., and Deutsch, M. (2001). Applying winwin to quality requirements: a case study. In *Proc. of ICSE*, pages 555–564.

In, H. P. and Olson, D. (2004). Requirements negotiation using multi-criteria preference analysis. *J.UCS*, 10(4):306–325.

In, H. P., Olson, D., and Rodgers, T. (2002). Multi-criteria preference analysis for systematic requirements negotiation. In *Proceedings of the Annual International Computer Software and Applications Conference (COMPSAC)*, pages 887–892.

Kazman, R., In, H. P., and Chen, H.-M. (2005). From requirements negotiation to software architecture decisions. *Information and Software Technology*, 47(8):511–520.

Khan, U. Z., Wahab, F., and Saeed, S. (2014). Integration of scrum with win-win requirements negotiation model. *Middle-East Journal of Scientific Research*, 19(1):101–104.

Kitapci, H. and Boehm, B. W. (2007). Formalizing informal stakeholder decisions–a hybrid method approach. In *Proc. of HICSS*.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University.

Le Bon, G. (1895). *Psychologie des foules*. UltraLetters.

Lee, M.-j. (1996). *Foundations of the winwin requirements negotiation system*. PhD thesis, University of Southern California.

Méndez Fernández, D., Wagner, S., Kalinowski, M., and et al. (2016). Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. *Empirical Software Engineering (doi:10.1007/s10664-016-9451-7)*, pages 1–41.

Raja, B. S., Iqbal, M. A., and Ihsan, I. (2007). Moving from problem space to solution space. *Int. J. of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 1(11):764–767.

Ramires, J., Antunes, P., and Respício, A. (2005). Software requirements negotiation using the software quality function deployment. In *International Conference on Collaboration and Technology (CRIWG)*, pages 308–324.

Ruhe, G. (2002). Software engineering decision support–a new paradigm for learning software organizations. In *Int. Workshop on Learning Software Organizations*, pages 104–113.

Ruhe, G., Eberlein, A., and Pfahl, D. (2002). Quantitative winwin: A new method for decision support in requirements negotiation. In *Proc. of SEKE*, SEKE '02, pages 159–166, New York, NY, USA. ACM.

Ruhe, G., Eberlein, A., and Pfahl, D. (2003). Trade-off analysis for requirements selection. *IJSEKE*, 13(04):345–366.

Schopenhauer, A. (1831). *Die Kunst zu beleidigen*. C.H. Beck.

Stallinger, F. and Grünbacher, P. (2001). System dynamics modelling and simulation of collaborative requirements engineering. *JSS*, 59(3):311 – 321.

Vogl, H., Lehner, K., Grünbacher, P., and Egyed, A. (2011). Reconciling requirements and architectures with the cbsp approach in an iphone app project. In *Proc. of RE*, pages 273–278.

Wieringa, R., Maiden, N., Mead, N., and Rolland, C. (2005). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requirements Engineering*, 11(1):102–107.