A Model for Evaluating Requirements Elicitation Techniques in Software Development Projects

Naiara C. Alflen[1][a], Edmir P. V. Prado[1][b] and Alexandre Grotta[1,2][c]
[1]*IS Post-graduation Program (PPgSI), University of São Paulo (USP), São Paulo, Brazil*
[2]*IS Graduation Program, Federal Institute of Sao Paulo (IFSP), São Paulo, Brazil*

Keywords: Requirements Elicitacion, Techniques, Experiment, Software Development.

Abstract: Requirements elicitation is the understanding of the real need of the user. It is considered the most complex and critical activity in software development. In the process of eliciting requirements several techniques are found in the literature. The main techniques described in the literature are presented in this article. Based on the techniques found in the literature, a model was proposed to analyze the influence and moderation of team involvement in eliciting requirements and the number of techniques used. The model was applied in an experimental format with students of the Information Systems course at the University of São Paulo. The results of the experiment are presented in this article.

1 INTRODUCTION

Requirements elicitation (RE) is one of the major software development activities and the most critical (Adikara et al., 2014; Romero et al., 2009; Sadiq et al., 2009; Vijayan et al., 2016). It is not just about writing requirements, but about discovering and understanding the real problem and user needs (Araujo et al., 2015).

Understanding software requirements can be considered one of the most difficult tasks of Requirements Engineering (Vijayan et al., 2016). According to Mishra et al., (2008), the major flaws in software projects are due to incomplete and incorrect requirements. Misunderstanding the user's need is one of the main failure factors of a project (Gonzales and Leroy, 2011). Most of the time, users have difficulty expressing their requirements (Nuseibeh and Easterbrook, 2000).

Scientific literature describes several RE techniques. In some of these techniques user participation is not addressed. However, in other techniques user participation is critical. In addition, Jayatilleke and Lai (2018) identified difficulties in the user engagement process, even using agile methodologies.

For Knauss et al., (2018), in today's changing organizational environment, software development methodologies need to evolve. Traditional Requirements Engineering is insufficient to support a more complex RE. Even though RE should be done at an early stage of software development (De Camargo Curcio et al., 2018), it does not adequately support changing requirements (Asghar et al., 2017). However, different approaches to project management influence this process. According to Asghar et al., (2017), Agile approach supports changing requirements better than waterfall approach, because in the first case the RE process occurs throughout project development.

According to Ali and Lai (2017) and Mishra et al., (2018) there are several techniques for RE in the literature. In addition, several authors describe the importance of user participation in RE (Castro-Herrera et al., 2009b; Seyff et al., 2010; Katina et al., 2014; De Angelis et al., 2018). This is because software success depends on the correct elicitation of the requirements (Babar et al., 2018). Hidalga et al., (2016) argue that user engagement and early problem identification improves software development, and the correct use of techniques is also required. But the correct use of techniques is not enough and the combination of various techniques improves the quality of requirements definition and improves understanding of the problem (Mishra et al., 2018; Alvertis et al., 2016).

The objective of this research is to develop a model to analyze the influence of team members in-

[a] https://orcid.org/0000-0001-9666-1417
[b] https://orcid.org/0000-0002-3505-6122
[c] https://orcid.org/0000-0003-2549-138X

Alflen, N., Prado, E. and Grotta, A.
A Model for Evaluating Requirements Elicitation Techniques in Software Development Projects.
DOI: 10.5220/0009397502420249
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 2*, pages 242-249
ISBN: 978-989-758-423-7

volvement and simultaneous use of RE techniques in the quality of functional requirements definition. We performed a systematic literature review that served as the basis for the model development.

2 METHOD AND MATERIALS

This is a qualitative study. According to Creswell (2007), qualitative research is useful for researchers to know more about research variables.

This section presents the research phases and the procedures to perform the model for evaluating RE techniques in software development projects.

2.1 Research Phases

The research steps are outlined in figure 1. Initially, we perform a Systematic Literature Review (SLR) to identify the most cited RE techniques. Based on this literature review we proposed a model for RE techniques assessment. Then, we perform an experiment with 53 students to improve de model.

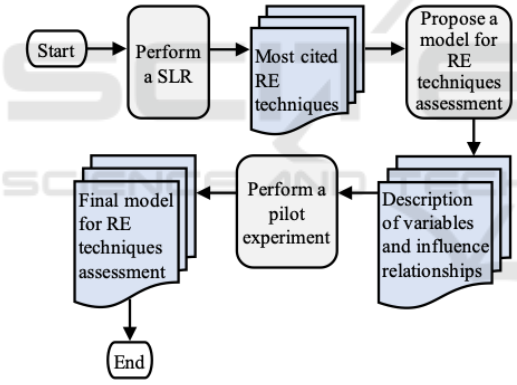

Figure 1: Research phases.

3 SYSTEMATIC LITERATURE REVIEW

The selection of articles was made in datasets related to the knowledge areas that address the research objective. We conducted a systematic literature review (SLR) based on the work of Kitchenham (2009). The datasets used are: (www.scopus.com); (http://ieeexplore.ieee.org); (https://acm.org).

We considered the period between 2012 and 2019, as it was desired to obtain the most updated works on the search subject. The literature search question was: What are the techniques used in the RE process in software development projects? To address this question, we defined the following search string: ("requirements elicitation" OR "requirements gathering") AND (techniques OR methods OR procedures OR tools OR artifacts OR specification) AND (software OR system OR systems).

3.1 Studies Selection Criteria

Based on the study by Kitchenham (2009), the protocol considered the following criteria for article selection:

(1) Inclusion criteria.

- Studies that had assessment or application of RE techniques in software development projects.
- Documents published between 2008 and February 2019.

(2) Exclusion criteria. In order to select only relevant papers, we excluded articles that:

- Are not related to Computer Science, Information Systems or Engineering
- Are not articles from journals or conferences.
- Were duplicate in the databases.
- Had not the full text available for consultation.

(3) Quality criteria. The risk analysis execution process was evaluated in the articles to verify that the procedure was fully described and was reproducible

3.2 Conducting the Review

Three steps were performed to select the articles. First, we search in the datasets and extracted 1789 articles. Then we applied the inclusion criteria and only 1351 articles were selected. Finally, we applied the exclusion criteria and 68 articles were selected. Figure 2 shows the three steps performed.

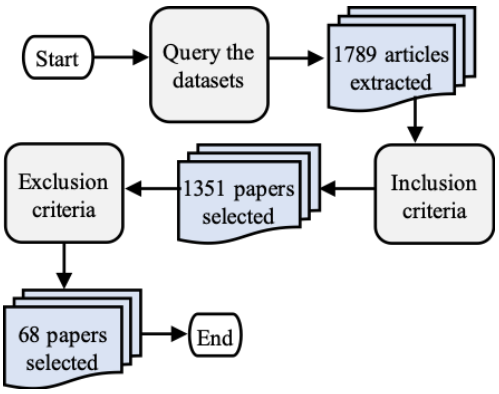

Figure 2: Studies selection process.

3.3 Reporting the Review

This section presents the RE techniques found in the literature. The techniques presented are from works related to software development. However, these same techniques are also found in articles from other fields of knowledge. Besides that, more than one technique for RE was found, in most of the works. In some articles the techniques are described and in other ones only cited.

- **Interview.** This was the most mentioned technique in the datasets. It was mentioned in 40 of the 68 selected articles. Interviews are critical for gathering information for new projects and domains (Carod and Cechich, 2010), and help in finding conflicts and policies. In interviews the information collected is more in-depth, however it involves a limited number of respondents (Saad and Dawson, 2018). Ilyas et al., (2017) state that the interview technique is the best one and the most used to define users' requirements. In interviews, success depends on the interviewer's ability to conduct the interview and collect the requirements (Gill et al., 2014).
- **Questionnaire.** It is a traditional RE technique (Abd Elmonem et al., 2018). However, there are difficulties in applying this technique. Questionnaires do not work well to analyze user experience (Fehlmann and Kranich, 2013). They often do not bring the expected results and should always be complemented with another technique (Carod and Cechich, 2010).
- **Use Cases.** This is the third most frequent technique amongst the selected articles. Use cases are the main instrument used to define requirements and communicate with users (Hajri et al., 2018). In addition, this technique is a starting point for identifying functional requirements (Ramesh and Reddy, 2016) and security requirements (Tondel et al., 2008; Li et al., 2017; Raspotnig et al., 2018).
- **Brainstorming.** This technique is performed in two steps. In the first, ideas are collected and in the second step discussions about them are held (Younas et al., 2017). Similarly Al-Qudah et al., (2013) state that the technique is used for data collection and further refining of ideas. However, the technique is not suitable for security RE, i.e., it does not result in a consistent set of security requirements (Sadiq et al., 2009). User experience should be considered for the correct use of this technique. Users who have experience with software development can better express their needs. In this case, techniques such as brainstorming are effective (Mishra et al., 2018).
- **Scenarios.** They help to discover the goals of the software. They are descriptions of current and future software processes contemplating user interaction (Adem and Kasirun, 2010).
- **Prototypes.** The prototype is an incomplete version of software that can be either disposable or scalable (Younas et al., 2017). The disposable prototype is used only to understand user requirements and perception. While the evolutionary prototype provides a basis for the final client version and may be usable. The prototype technique facilitates user engagement and early identification of issues that may affect software usability (de la Hidalga et al., 2016; Ramakrishnan et al., 2014). A disadvantage of the technique is the waste of time and resources (Gil, 2008).
- **Focus Groups.** It is a technique that allows knowing the user's wishes and perceptions regarding the software, as well as the definition of the requirements (Younas et al., 2017). It is similar to an interview, however with the involvement of a group (Pitula and Radhakrishnan, 2011). With the application of this technique it is possible to collect different opinions (Alvertis et al., 2016). However, there is the disadvantage of participants feeling uncomfortable when stating opinions different from those stated by the group (Pitula and Radhakrishnan, 2011). Another problem is dominant and biased participants (Fernandes et al., 2012), that do not explore valid ideas from other participants (Pitula and Radhakrishnan, 2011).
- **Workshop.** It is another collaborative technique for defining the requirements of a system (De Angelis et al., 2018). It can be used to clarify ambiguities (Mishra et al., 2018). Because it is a collaborative technique, it has the same problem as focus groups: dominant and biased participants (Fernandes et al., 2012).
- **Joint Application Development (JAD).** According to researchers Sadiq et al., (2009) JAD's goal is to involve stakeholders in the software development process through structured meetings. Meetings include a facilitator, product end users, developers, and observers. During the sessions the RE team is responsible for identifying and collecting the information. In the last JAD session the requirements are validated by the stakeholders. There is a disadvantage to this technique: if there are many JAD sessions, users may feel that developers are shifting their responsibilities to them. The technique also has the difficulty of previous

collaborative techniques - dominant and biased participants as it brings together a wide range of stakeholders (Castro-Herrera et al., 2009a).

- **User Stories.** User stories are used with agile methodologies (Babar et al., 2018; Younas et al., 2017). They enable and help users to document their own needs (Seyff et al., 2010). They are brief descriptions of the software's functionality (Mobasher and Cleland-Huang, 2011; Younas et al., 2017). User stories are discussed during all stages of development to clarify requirements (Knauss et al., 2018).

3.4 Related Works

In SLR two empirical works were found to select the most used techniques. From these works and techniques found in SLR, the techniques for model development were selected.

The first work is by authors Kassab et al., (2014) who presented the most applied techniques in the business context. The study on requirements elicitation techniques was conducted in 2013 (Figure 3). The authors conducted a survey of 247 participants from 23 countries where 119 completed the survey to understand the techniques used to elicit requirements.

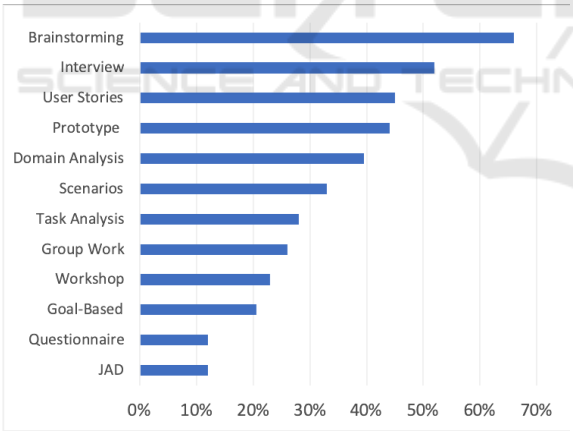

Figure 3: Kassab et al., (2014) research.

As a result, 65% of respondents used the brainstorming technique for elicitation and in the second position the interview technique. However in this SLR the brainstorming technique was cited in 21 documents while interviews in 40 documents.

The user stories technique was presented in third position with approximately 45%. While in this SLR the user stories technique was in tenth position. With less difference was the prototype technique occupying the fourth position of the survey and the fifth position in this SLR. The domain analysis technique present in approximately 38% of interviewed companies had low occurrence in this SLR found in only one article in the context of software development. The scenarios technique maintained a similar position in both surveys. The survey was presented in sixth position with 34% of the companies and in this SLR is presented in fifth position in 16 articles. Task analysis and group work techniques with a little over 25% survey presence also had low influence on this SLR found in just one article.

Yet in the survey by Kassab, Neill and Laplante (2014) the questionnaires technique is presented with low occurrence along with the JAD technique with just over 10% of use. However, in this SLR the questionnaires technique was the second most cited present in 28 articles. And the JAD technique was found in 11 articles occupying the ninth position. The other techniques presented by the authors also had low occurrence in this SLR and thus will not be compared.

The second work is by Todoran et al., (2013). The authors conducted an exploratory study with 19 companies from 10 countries to understand the elicitation of requirements in the cloud context. The study was conducted through interviews with the main objective of identifying the techniques for eliciting requirements used by cloud providers. Part of the study is presented in figure 4.

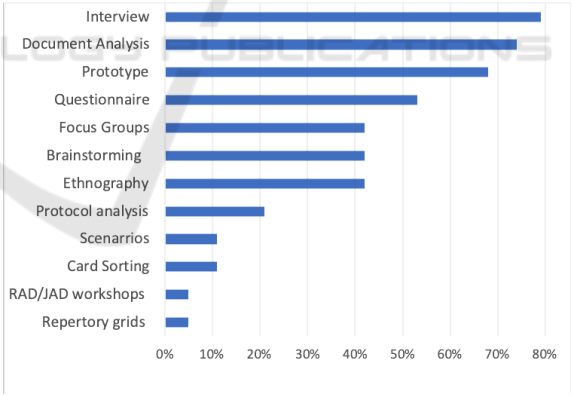

Figure 4: Todoran et al., (2013) research.

Although the cloud context was approached by the authors as something new, the research result shows that the traditional approaches to requirements gathering are the most applied. Interviews and document analysis were the main techniques identified and applied in over 70% of companies. Interviews were the main technique applied following the same results of this SLR. The document analysis technique with strong application in the study was visible in only 9 articles of this SLR.

The prototype technique was presented in more

than 60% of the companies. This SLR was found in 16 documents along with the technical scenarios. However in the research by Todoran et al., (2013) the scenario technique had low application in the interviewed companies with approximately 10% of adherence.

In this SLR the questionnaires technique was the second most cited. However in the work of the authors (Todoran et al., 2013) appears only in the fourth position. Another technique with good placement in this SLR was the brainstorming present in 21 documents while in the author's research it appears in less than 40% of the companies. JAD / RAD techniques also appear to have poor adherence in the work of authors with less than 10% adherence. However in this SLR the JAD appears in 11 articles. The other techniques present in the author's work have low adherence in the companies surveyed and also in this SLR.

In the SLR of Dar et al., (2018), 18 primary studies were found for the selection of requirements elicitation techniques. The author's found results similar to this review: interview 72%, prototypes 44%, user stories 44%, questionnaries 33% and brainstorming 27%. The review (Dar et al., 2018) is not present in the SLR of this work.

From the combination of these two articles and the SLR, the following techniques were selected for study: interviews, braisntorming, prototypes and user stories. The selected techniques are as the first 4 techniques present in the work of Kassab et al., (2014).

4 STEPS TO PERFORM THE EXPERIMENT

The proposal for the pilot experiment was the development of an employee evaluation management software developed by the undergraduate students in Information Systems at the University of São Paulo (USP).

The students of the USP Masters and Doctorate in Information Systems Program defined the software requirements according to their needs. The requirements were set by the students through brainstorming sessions. The list of requirements established in these sessions is available in https://docs.google.com/document/d/1SbtqLi3P88Xje _M-RoQeJ59W3EAT9etHMtBr8TnRj_o/ edit? usp = sharing. After the software development by undergraduate students, the master students undertook the assessments.

Development was carried out by project groups. The class was divided into 12 groups with 4 members and 2 groups with 3 members. For the experiment, the groups were divided between test and control. In the test groups there was the interaction of all team members with the researcher and the application of the techniques for elicitation: interviews, brainstorming, user stories and prototypes. In the control group the interaction happened only with one team member and with the techniques: interviews, user stories and prototypes. The interaction with the researcher was held in 4 meetings to perform Sprints. In the second, third and fourth interaction the groups presented a prototype of the software. At the end of the 4 sprints all groups showed a functional prototype of the software.

After the software development was completed, the master students performed the tests on the software prototypes. In the testing process the list of requirements established in the first step was used for evaluation.

All students of the master evaluated all the software developed. The evaluations occurred without knowledge of the type of group used in each software. The evaluations were applied through a survey. Survey questions were conducted within the context of functional and non-functional requirements with a Likert scale of 1 to 5.

The survey was applied through Google Forms available at: https://forms.gle/59hQVFcebedKnAz 49.

Table 1: Survey average.

Group	Type	Average
Group 04	Test	10
Group 03	Test	9,63
Group 14	Test	9
Group 05	Test	8,55
Group 11	Test	8,28
Group 09	Test	7,2
Group 07	Test	6,48
Group 13	Control	9,03
Group 06	Control	8,73
Group 02	Control	8,53
Group 08	Control	8,05
Group 01	Control	7,93
Group 12	Control	7,5
Group 10	Control	4,28

The application resulted in 10 evaluations of each software developed. Thus, 140 evaluation responses were analyzed and compared. The strategy for processing these data was to separate the evaluations into two groups:

- In the first group, the answers to the questions regarding the functional requirements defined in

the first stage of the experiment were evaluated. These questions are divided to analyze the 5 blocks of functional requirements: performance, behavior, personal skills, social skills, legal and ethical aspects.

- In the second group, the answers to question regarding non-functional requirements: privacy, usability and portability were evaluated.

In each group the scores of the questions were added and the arithmetic mean was performed. Thus, was found the result available in the table 1.

5 PROPOSED MODEL

From the 14 projects developed and their results, the model available in figure 5 was defined. Two guiding questions for analysis were defined, the team involvement and the number of techniques used.

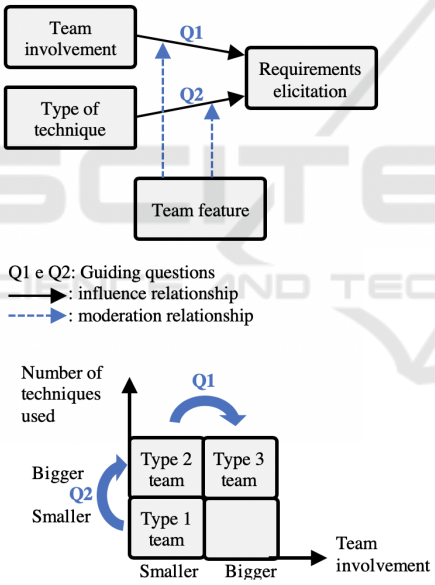

Figure 5: Proposed model.

In the application of the model will be compared the use of only one technique for requirements elicitation with the combination of techniques for requirements elicitation. The technique defined for the groups of only one technique was the interview. In groups with the combination of techniques will be used these ones: interview, brainstorming and user stories. Another factor to be analyzed is the team's participation in data collection. In some groups all members will participate in data collection, in other groups only one member. The group types will be:

- **Type 1:** Use of a single technique with partial team participation in data collection;
- **Type 2:** Use of combined techniques with partial team participation in data collection;
- **Type 3:** Use of combined techniques and participation of all staff in data collection.

This model will be applied in the first half of 2020 and for data analysis, the two variables will be considered: involvement and techniques.

At first moment, all groups will be evaluated for their involvement in the project. Involvement will be assessed by team participation and communication. These data will be captured by the researcher's communication with everyone involved even in teams with partial participation. In addition to the researcher's perception of direct contact with those involved, anonymous and group surveys will be carried out, so that everyone can express their opinions about communication and team participation in the development of the project's activities.

In the second moment, all groups will be evaluated by the techniques used. The technical variable will be evaluated by the amount of techniques used and the application of the techniques. For quantity analysis, the final results of each project will be compared between the single and combined techiniques teams. The data for evaluating the application of the techniques will be obtained from the researcher's communication with those involved in the interactions and from the material delivered by the teams, such as user stories.

The communication, participation and application fields will be evaluated on a scale of 1 to 3 (1 - excellent, 2 - good, 3 - regular).

6 EXPRIMENT OUTCOMES

During the experiment, student's behavior during interactions was observed. In the behavior was analyzed the communication and interaction between the members of the test teams. In the control teams, the position of the responsible for the team was observed. Interviews and confidential questionnaires were conducted in all teams to analyze the perceptions of the researcher and team members regarding the progress of the project.

In order to compose the teams evaluation, during the sprints the user stories delivery and presentation of the software prototypes were requested. At the end of the sprints the final delivery of the project was made.

From the data triangulation: observation, documentation with the evaluations and interviews it was possible to observe:

- The correct use of the techniques influences the software development. In the groups with an average below 8 (Test: 9 and 7; Control 1, 12 and 10) all made low-fidelity user stories. The delivered documents had only 3 or 4 user stories, while on other teams the documents had an average of 15 stories. Two groups (10 and 12) delivered the user stories only in the last sprint.
- Teams with a dominant member make it difficult to understand the requirements. In group 7 the team had difficulty describing user stories due to lack of integration between members and the superiority of one of the members. This team member would not allow anyone else to express their ideas during requirements elicitation and prototype presentation.

These aspects were mentioned by other authors:

- Mishra et al., (2018), Ali and Lai (2017), Elsaid et al., (2016) present the need for the correct use of techniques in their context and application. Other authors mention that the success of the techniques also depends on the engagement among the participants (Pinto et al., 2017).
- Pitula and Radhakrishnan (2011) and Fernandes et al., (2012) present that dominant and biased participants leave valid ideas of other participants unexplored.

7 CONCLUSIONS

The presented experiment allowed the evaluation and improvement of the proposed model. This model will be applied to a new experiment in the first half of 2020. The model will be applied to software development organizations.

During the experiment some situations that impact on projects were found. The behavior of team members directly impacts project completion. Teams with low communication and conflicts between members had a lower performance than the others.

Proper use of the techniques is another important factor. Teams in which the techniques were not applied at the beginning of the project and without seriousness in the application had lower performance. The application of the model in the educational environment is the main limitation of this experiment.

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor and PPgSI for support in the participation of ICEIS.

REFERENCES

Abd Elmonem, M., Nasr, E., and Gheith, M. (2018). Automating requirements elicitation of cloud-based erps. *Advances in Intelligent Systems and Computing*, 639:171–180.

Adem, N. and Kasirun, Z. (2010). Automating function points analysis based on functional and non functional requirements text. volume 5, pages 664–669, Singapore.

Adikara, F., Hendradjaya, B., and Sitohang, B. (2014). A new proposal for the integration of key performance indicators to requirements elicitation process originating from organization goals. Institute of Electrical and Electronics Engineers Inc.

Alvertis, I., Papaspyros, D., Koussouris, S., Mouzakitis, S., and Askounis, D. (2016). Using crowdsourced and anonymized personas in the requirements elicitation and software development phases of software engineering. pages 851–856. Institute of Electrical and Electronics Engineers Inc.

Araujo, R., Anjos, E., and Silva, D. (2015). Trends in the use of design thinking for embedded systems. pages 82–86.

Asghar, A., Tabassum, A., Bhatti, S., and Jadi, A. (2017). Impact and challenges of requirements elicitation & prioritization in quality to agile process: Scrum as a case scenario. pages 50–55. Institute of Electrical and Electronics Engineers Inc.

Babar, A., Bunker, D., and Gill, A. (2018). Investigating the relationship between business analysts' competency and is requirements elicitation: A thematic-analysis approach. *Communications of the Association for Information Systems*, 42(1):334–362.

Carod, N. and Cechich, A. (2010). Cognitive profiles in understanding and prioritizing requirements: A case study. pages 341–346, Nice.

Castro-Herrera, C., Cleland-Huang, J., and Mobasher, B. (2009a). Enhancing stakeholder profiles to improve recommendations in online requirements elicitation. pages 37–46, Atlanta, GA.

Castro-Herrera, C., Duan, C., Cleland-Huang, J., and Mobasher, B. (2009b). A recommender system for requirements elicitation in large-scale software projects. pages 1419–1426, Honolulu, HI.

Dar, H., Lali, M. I., Ashraf, H., Ramzan, M., Amjad, T., and Shahzad, B. (2018). A systematic study on software requirements elicitation techniques and its challenges in mobile application development. *IEEE Access*, 6:63859–63867.

De Angelis, G., Ferrari, A., Gnesi, S., and Polini, A. (2018). Requirements elicitation and refinement in collaborative research projects. *Journal of Software: Evolution and Process*, 30(12).

De Camargo Curcio, K., Navarro, T., Malucelli, A., and Reinehr, S. (2018). Requirements engineering: a systematic mapping study in agile software development. 139.

de la Hidalga, A., Hardisty, A., and Jones, A. (2016). Scram–CK: applying a collaborative requirements engineering process for designing a web based e-science toolkit. *Requirements Engineering*, 21(1):107–129.

Fehlmann, T. and Kranich, E. (2013). Customer-driven software product development software products for the social media world? a case study. *Communications in Computer and Information Science*, 364 CCIS:300–312.

Fernandes, J., Duarte, D., Ribeiro, C., Farinha, C., Pereira, J., and Da Silva, M. (2012). Ithink : A game-based approach towards improving collaboration and participation in requirement elicitation. volume 15, pages 66–77, Genoa. Elsevier B.V.

Gil, A. C. (2008). *Métodos e técnicas de pesquisa social*. Atlas SA, São Paulo, Brasil.

Gill, K., Zaidi, A., and Kiani, M. (2014). Eliciting futuristic end-user requirements through contributory appreciative inquiry. pages 49–54. Institute of Electrical and Electronics Engineers Inc.

Gonzales, C. and Leroy, G. (2011). Eliciting user requirements using appreciative inquiry. *Empirical Software Engineering*, 16(6):733–772.

Hajri, I., Goknil, A., Briand, L., and Stephany, T. (2018). Configuring use case models in product families. *Software and Systems Modeling*, 17(3):939–971.

Katina, P., Keating, C., and Jaradat, R. (2014). System requirements engineering in complex situations. *Requirements Engineering*, 19(1):45–62.

Knauss, E., Yussuf, A., Blincoe, K., Damian, D., and Knauss, A. (2018). Continuous clarification and emergent requirements flows in open-commercial software ecosystems. *Requirements Engineering*, 23(1):97–117.

Li, H., Li, X., Hao, J., Xu, G., Feng, Z., and Xie, X. (2017). Fesr: A framework for eliciting security requirements based on integration of common criteria and weakness detection formal model. pages 352–363. Institute of Electrical and Electronics Engineers Inc.

Mishra, D., Aydin, S., Mishra, A., and Ostrovska, S. (2018). Knowledge management in requirement elicitation: Situational methods view. *Computer Standards and Interfaces*, 56:49–61.

Mobasher, B. and Cleland-Huang, J. (2011). Recommender systems in requirements engineering. *AI Magazine*, 32(3):81–89.

Nuseibeh, B. and Easterbrook, S. (2000). Requirements engineering: A roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 35–46, New York, NY, USA. ACM.

Pinto, R., Silva, L., Lucena, M., and Aleixo, F. A. (2017). A software process line for combinational creativity-based requirements elicitation. In *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 2: ICEIS,*, pages 360–369. INSTICC, SciTePress.

Pitula, K. and Radhakrishnan, T. (2011). On eliciting requirements from end-users in the ict4d domain. *Requirements Engineering*, 16(4):323–351.

Ramakrishnan, L., Poon, S., Hendrix, V., Gunter, D., Pastorello, G., and Agarwal, D. (2014). Experiences with user-centered design for the tigres workflow api. volume 1, pages 290–297. Institute of Electrical and Electronics Engineers Inc.

Ramesh, R. M. and Reddy, S. C. (2016). A survey on security requirement elicitation methods: Classification, merits and demerits. *International Journal of Applied Engineering Research*, 11(1):64–70.

Raspotnig, C., Karpati, P., and Opdahl, A. (2018). Combined assessment of software safety and security requirements: An industrial evaluation of the chassis method. *Journal of Cases on Information Technology*, 20(1):46–69.

Romero, M., Vizcaíno, A., and Piattini, M. (2009). Teaching requirements elicitation within the context of global software development. In *2009 Mexican International Conference on Computer Science*, pages 232–239.

Saad, A. and Dawson, C. (2018). Requirement elicitation techniques for an improved case based lesson planning system. *Journal of Systems and Information Technology*, 20(1):19–32.

Sadiq, M., Ghafir, S., and Shahid, M. (2009). An approach for eliciting software requirements and its prioritization using analytic hierarchy process. pages 790–795, Kottayam, Kerala.

Seyff, N., Graf, F., and Maiden, N. (2010). Using mobile re tools to give end-users their own voice. pages 37–46, Sydney, NSW.

Todoran, I., Seyff, N., and Glinz, M. (2013). How cloud providers elicit consumer requirements: An exploratory study of nineteen companies. pages 105–114, Rio de Janeiro. IEEE Computer Society.

Tondel, I., Jaatun, M., and Meland, P. (2008). Security requirements for the rest of us: A survey. *IEEE Software*, 25(1):20–27.

Vijayan, J., Raju, G., and Joseph, M. (2016). Collaborative requirements elicitation using elicitation tool for small projects. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 340–344.

Younas, M., Jawawi, D., Ghani, I., and Kazmi, R. (2017). Non-functional requirements elicitation guideline for agile methods. *Journal of Telecommunication, Electronic and Computer Engineering*, 9:137–142.