The Journal of Systems and Software 139 (2018) 32–50

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Requirements engineering: A systematic mapping study in agile software development

Karina Curcio, Tiago Navarro, Andreia Malucelli, Sheila Reinehr*

Graduate Program in Computer Science (PPGIa), Pontifícia Universidade Católica do Paraná – PUCPR, Curitiba, Brazil

ARTICLE INFO

Article history:
Received 19 December 2016
Revised 16 December 2017
Accepted 24 January 2018
Available online 31 January 2018

Keywords:
Agile software development
Requirements engineering
Systematic mapping study

ABSTRACT

Context: Requirements engineering in agile software development is a relatively recent software engineering topic and it is not completely explored and understood. The understanding of how this process works on agile world needs a deeper analysis.
Objective: The goal of this paper is to map the subject area of requirements engineering in agile context to identify the main topics that have been researched and to identify gaps to develop future researches. It is also intended to identify the obstacles that practitioners face when using agile requirements engineering.
Method: A systematic mapping study was conducted and as a result 2171 papers were initially identified and further narrowed to 104 by applying exclusion criteria and analysis.
Conclusion: After completing the classification and the analysis of the selected studies it was possible to identify 15 areas (13 based on SWEBOK) where researches were developed. Five of such areas points to the need of future researches, among them are requirements elicitation, change management, measuring requirements, software requirements tools and comparative studies between traditional and agile requirements. In this research, some obstacles that practitioners face dealing with requirements engineering in agile context were also identified. They are related to environment, people and resources.

1. Introduction

Since the agile manifesto was released a lot of research has been intensively developed to explore the agile software development (Dybå and Dingsøyr, 2008; Diebold and Dahlem, 2014). The agile manifesto stated some values like *"individual and interactions over process and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan"* (Agile Manifesto: Manifesto for Agile Software Development 2001).

All of these values add flexibility and consequently agility to the software development process. However, recent studies indicate high rates of projects failure, including those that are using agile processes. According to a study published by the Standish Group (2014) about the results of software projects, five of the eight top projects cancellation factors are related to requirements. Incomplete requirements, low customer involvement, unrealistic expectations, changes in requirements and unnecessary requirements were listed as the main factors.

So, the question that remains is how to deal with it? The answer should be inside of requirements engineering area. In traditional requirements engineering some sequential activities are developed during five specific phases: elicitation, analysis, documentation, validation and verification phases (Kotonya and Sommerville, 1997). During all these phases it is necessary to manage the requirements. What is already known is that in the traditional development life cycle, all of these activities are done during the analysis phase of the software development life cycle. It is easy to imagine this occurring when a plan-driven approach is adopted, but when an agile approach is adopted all activities and phases described on traditional requirements engineering are not so clear. Agile requirements engineering (RE) activities are not sequential but are iterative and are performed during each of the several short development cycles [S1]. Requirements engineering in agile development is informal and based on the skills and knowledge of individuals (Dingsøyr et al., 2012). It is difficult to describe or characterize the agile requirements engineering because it is still cloudy, not only for software developers but for the research community too. The software development community as a whole is still unfamiliar with the role of the requirements engineering practices in agile methods (Inayat et al., 2015).

* Corresponding author.
E-mail addresses: malu@ppgia.pucpr.br (A. Malucelli), sheila.reinehr@pucpr.br (S. Reinehr).

https://doi.org/10.1016/j.jss.2018.01.036

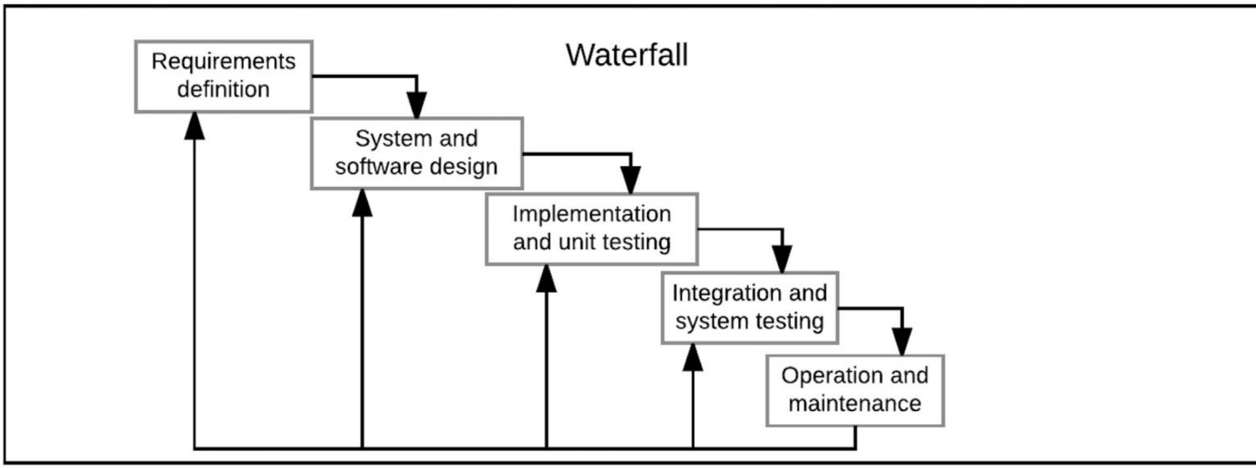

Fig. 1. Waterfall model based on Sommerville (2001).

Motivated by this perception, the goal of this study is to explore the agile requirements engineering and bring to light more discussions promoted by literature foundations. To attend this goal it is intended to create a clear and objective way of visualizing the results instigating those who intend to follow this line of research. At the beginning of the research it was noticed that the field of this research is very broad and we need to find a way to promote adequate visibility of the results, focusing on the extensions of the areas that the agile requirements engineering can achieve rather than its depth. As described by Kitchenham and Charters (2007) a systematic mapping study allows the evidence in a domain to be plotted at a high level of granularity. Systematic mapping studies or scoping studies are designed to give an overview of a research area through classification and counting contributions in relation to the categories of that classification (Petersen et al., 2008). It provides a structure of the type of research reports and results that have been published by categorizing them. It often gives a visual summary, the map, of its results (Petersen et al., 2008). To make the results emerge correctly and to reduce the bias of this research a systematic mapping methodology was applied. The research questions that we aimed to answer are the following:

1) On which requirements engineering topics are the researches on requirements engineering in agile software development concentrated?
2) What are the gaps concerning the requirements engineering in the context of agile software development?
3) What obstacles do the agile requirements engineering is facing (environment, people and resources)?

2. Theoretical background

Requirements engineering is concerned with identifying, modeling, communicating and documenting the requirements of a system and the context in which the system will be used [S2]. The use of the term “engineering” implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent and relevant (Kotonya and Sommerville, 1997). A requirement engineering process is a structural set of activities, which are followed to derive, validate, and maintain a system requirements document. In this section we will provide the theoretical background about requirements engineering in order to have a better understanding of the research context.

2.1. Traditional requirements engineering process

The term “traditional requirements engineering” is based on the waterfall life cycle model, which emerged in 1970s. In this approach all process for developing a system are executed in a sequential order, in which progress is seen as flowing steadily downwards through the phases of: requirements definition, system and software design, implementation and unit testing, integration and system testing and operation and maintenance (Sommerville, 2001). The representation of this model can be seen in Fig. 1.

In traditional requirements engineering a number of processes for gathering requirements in accordance with the needs and demands of the users are involved and all of them are executed during the requirements definition phase. The process begins with the elicitation activity where requirements and the system boundaries are discovered through the stakeholders. In this phase many techniques can be used, such as prototyping, brainstorming, interviews and use cases [S3]. Then the requirements analysis and negotiation activities start to get a better understanding of the whole business and to check if the elicited requirements are consistent, complete and feasible. Sometimes, during these activities, the requirements can be modeled to make them clearer for the developers. It is also possible to prioritize the requirements to satisfy some limitations such as time, resources or technical capabilities.

In the documentation activity the requirements are written and become a baseline for specifying all types of functional and non-functional requirements. The next activity in the sequence is the validation. The validation checks if the requirements statements are consistent and if they satisfy costumer’s needs. Test cases are also used in this phase to reveal the ambiguities and vagueness in written requirements [S3]. To support these activities a requirements management process should be introduced to manage changes to the requirements during all prior presented phases.

In waterfall approach it is normal to freeze parts of the development such as the specification (Sommerville, 2001). Problems are left for later resolution, ignored or programmed around and it may bring some problems like badly structured systems.

2.2. Agile requirements engineering

Differently from the traditional requirements engineering the term “agile requirements engineering” is recent and emerged from the agile manifesto in 2001, as the agile software development started to be explored. During the agile manifesto twelve principles were stated and one of them was directly related to requirements:

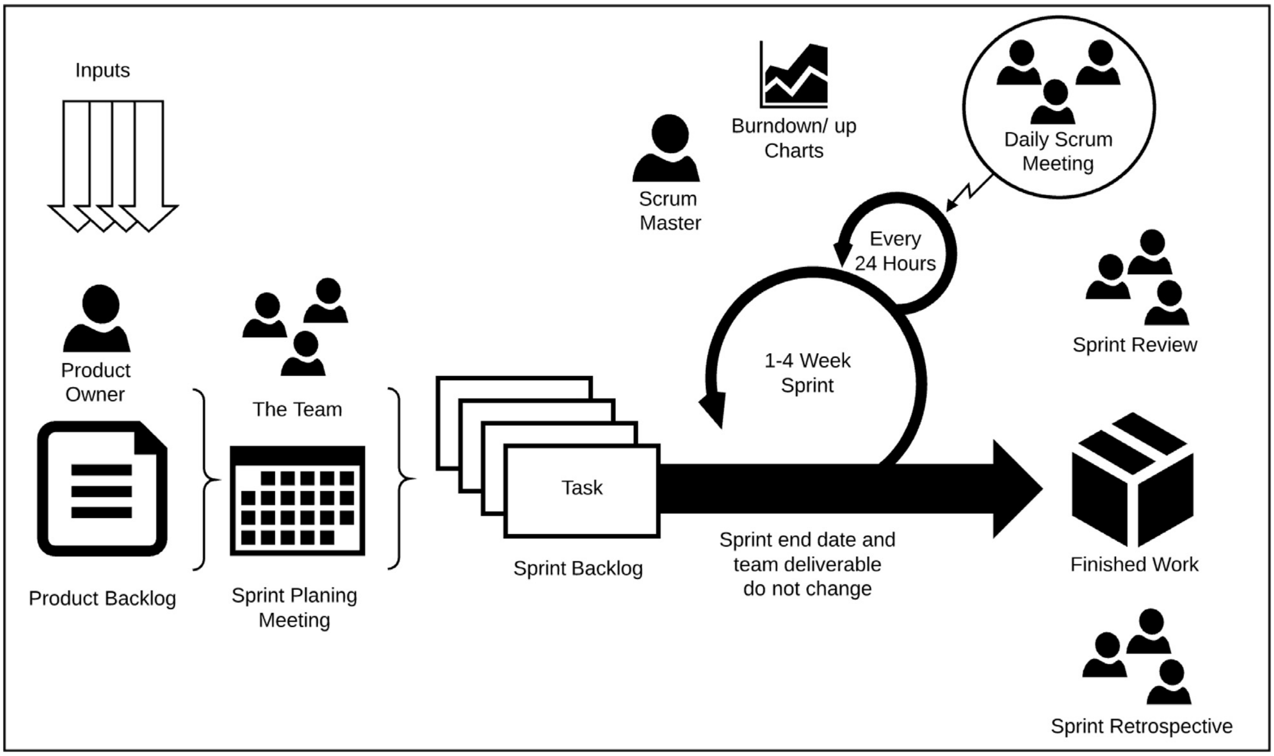

Fig. 2. Scrum Framework based on Schwaber and Sutherland (2018).

"Changes in requirements are welcome, even late in development." So it assumes that requirements engineering continues through the lifetime of a system. As the agile software development usually works with small iterations with frequent deliverables, the development process is dynamic. The requirements are initially defined with the customer and listed in a customer wish list format; every couple of weeks they are discussed (e.g. in the Scrum Framework illustrated in Fig. 2), better understood, and reprioritized, to define the scope of the next iteration [S4]. This new way of work has brought some questions in the community about how to deal with requirements engineering in such flexible and dynamic way of work. The role of the requirements engineering process is still unknown and it poses some new challenges to their execution.

Traditional requirements engineering has been explored in many researches in recent years and the interest in developing Systematic Literature Reviews (SLR) among requirements engineering has also been increasing (Bano et al., 2014). Bano, Zowghi and Ikram showed this in a tertiary study to provide a comprehensive overview of these published SLR in requirements engineering. In their study 53 distinct SLR published from 2006 and 2014 were identified.

As one of the goals of this paper is to map the subject area of requirements engineering in agile context to identify the main topics that have been studied and also to identify gaps in the literature to develop future researches, the need of identifying the main researches, using SLR and mapping studies emerged.

3. Related studies

During this research studies related to systematic reviews or mapping studies of requirements engineering in agile software development were found, as described in Table 1.

Inayat et al. (2015) conducted a systematic literature review on agile requirements engineering published between 2002 and June of 2013. In this study 21 papers were identified and the review identified 17 practices of agile requirements engineering, 5 challenges traceable to traditional requirements engineering that are resolved by the agile RE and 8 practical challenges posed by the practice of agile RE. Their findings suggest that agile requirements engineering as a research context needs additional attention and more empirical researches are required to better understand the impact of agile requirements engineering practices.

Medeiros et al. (2015) conducted a mapping study on requirements engineering in agile projects based in evidences of industry in 2014. In this study 24 papers were identified but they did not consider papers published in 2014 because the research was under way. The goal of this study was to conduct an exploratory study to investigate how requirements engineering is used in projects that adopt agile methodologies based on evidences of industry. Their findings show that the low user's involvement and the constant change of requirements were identified as the main challenges to be overcome.

Heikkila et al. (2015) conducted a mapping study on requirements engineering in agile on September of 2014. In this study 28 papers were identified and analyzed. Their findings show that the definition of agile RE is vague. The result also shows some benefits and some problematic areas of agile RE. The authors also reported some solutions to the identified problems.

Schön et al. (2016) conducted a systematic literature review between 2007 and 2015. The goal of this study was to capture the current state of the art of the literature related to Agile RE with focus on stakeholder and user involvement. In this study 27 relevant papers were identified with an extensive quality assessment of the included studies. After analyzing them in detail the authors derive deep insights to the following aspects of Agile RE: stakeholder and user involvement, data gathering, user perspective, integrated methodologies, shared understanding, artifacts, documentation and Non-Functional Requirements (NFR).

Eghariani and Kama (2016) conducted a systematic literature review between January 2000 and June 2015. The aim of this study was to fill the gap by presenting requirements engineering prac-

Table 1
Systematic reviews and mapping studies on requirements engineering in agile software development.

Authors	Goal	Research questions
Inayat et al. (2015)	Conducted a systematic review of the literature to map the evidence available about requirements engineering practices adopted and challenges faced by agile teams in order to understand how traditional requirements engineering issues are resolved using agile requirements engineering.	(RQ1) What are the adopted practices of agile RE according to published empirical studies? (RQ2) What are the challenges of traditional RE that are resolved by agile RE? (RQ3) What are the practical challenges of agile RE?
Medeiros et al. (2015)	Conducted a systematic mapping study to investigate how requirements engineering is used in projects that adopt agile methodologies based on evidences of Industry.	*Principal Research Question:* (PRQ) How the requirements engineering has been conducted in projects that adopt agile methodologies? *Specific Research Questions:* (SRQ1) In order to elicit requirements, which requirements engineering techniques are being used in projects that adopt agile methodologies? (SRQ2) In order to specify requirements, which requirements engineering techniques are being used in projects that adopt agile methodologies? (SRQ3) What are the challenges and limitations of Requirements Engineering techniques used in agile projects? (SRQ4): What are the implications for the software industry and academia, reported in the current studies involving the Requirements Engineering in Agile projects?
Heikkila et al. (2015)	Conducted a systematic mapping study to have a better understanding of requirements engineering in agile software development.	(RQ1) What has been researched regarding requirements engineering in an agile context? (RQ2) What are the reported key benefits of agile requirements engineering? (RQ3) What are the reported problems and corresponding solutions related to agile requirements engineering?
Schön et al. (2016)	Conducted a systematic review to capture the current state of the art of the literature related to Agile RE with focus on stakeholder and user involvement. In particular investigate what approaches exist to involve stakeholder in the process, which methodologies are commonly used to present the user perspective and how requirements management is been carried out.	(RQ1) What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD? (RQ2) Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found? (RQ3): What are the common ways for requirements management in ASD?
Eghariani and Kama (2016)	Conducted a systematic review to fill the gap by presenting requirements engineering practices in agile methodology and also the challenges of requirements engineering activities, which are faced by agile team members.	(RQ1) What are the agile requirements engineering practices?
Heck and Zaidman (2016)	Conducted a systematic review to investigate what quality criteria for assessing the correctness of written agile requirements exists.	(RQ2) What are agile methodology requirements engineering challenges?
Medeiros et al. (2016)	Conducted a systematic mapping study to investigate the phenomenon of the quality on software requirements specification in agile software development and build an explanatory model about it.	(RQ1) Which are the known quality criteria for agile requirements specification?
Magües et al. (2016)	Conducted a systematic mapping study to investigate de integration of the agile software development and user-centred design.	(RQ1) How is the quality of software requirements specification affected in agile software development?
Silva et al. (2011)	Conducted a systematic review of existing literature on the integration of agile software development with user-centered design approaches.	(RQ1) What is the current state of the integration of agile processes and usability? (RQ1) How are usability issues addressed in Agile projects? (RQ2) What are common practices to address usability issues in Agile Methods?

tices in agile methodologies and also the challenges of requirements engineering activities, which are faced by agile team members. In this study 22 relevant research papers were identified and through them it was possible to determine 16 of agile requirements engineering activities, also 6 challenges of requirements engineering in agile methodologies.

Heck and Zaidman (2016) conducted a systematic literature review between 2001 and 2014. The goal of this study was to investigate what quality criteria for assessing the correctness of written agile requirements exists, as the quality of requirements is typically considered as an important factor for the quality of the end product. In this study 16 selected papers were identified and 28 different quality criteria for agile requirements specification were described. After analyzing and categorizing them a comparison was made using the criteria from traditional requirements engineering. The authors also discussed the findings in the form of recommendations for practitioners on quality assessment of agile requirements.

Medeiros et al. (2016) conducted a systematic mapping study to investigate the phenomenon of the quality on Software Requirements Specification (SRS) in Agile Software Development and built an explanatory model about it. They used a mixed method research strategy for creating a rich description of the factors that affect the quality using a systematic mapping study and a cross-case analysis of two software organizations.

Magües et al. (2016) conducted a mapping study to investigate the integration of the agile software development and user-centered design. The review covered papers published up until 15 October 2015. In this study a total of 161 primary studies were retrieved, which were categorized according to four criteria: process, practice, team and technology integration. After analyzing them the authors reported the current state of usability in the agile software development process.

Silva et al. (2011) conducted a systematic review of existing literature in August of 2010. The aim of this study was to present the results of the integration of agile methods and user-centered design approaches. After the initials filtering, 58 papers were selected for further evaluation. The studies were classified regarding their content, research method and considering the techniques used by the teams that were studies. These issues were used as the basis for a proposal of a process model of software development combining UCD with Agile principles.

Despite of Heikkila et al. (2015) has already developed a mapping study in the same subject area of this study, it was still needed to explore this subject in a deeper manner. The reason for this decision is because it was not possible to identify in their work what are the less explored or not completed explored topics regarding to requirements engineering in agile context. In their study some research topics regarding to requirements engineering in agile context were only pointed out as presented topics in actual research scenario. Practices, communications, requirements prioritization and comparative studies (traditional RE and agile RE) were some of them. But it was not clear what are the gaps related to requirements engineering in agile context. So, the intention of this work is to identify on which requirements engineering topics are concentrated the researches on requirements engineering in agile software development, what are the gaps concerning the topic and what are the obstacles the practitioner of agile requirements engineering will face. We have decided to include all studies that were found in our research despite of being empirically evaluated or not, in order to have a broader view of the concentration of the requirements engineering topics as many of them were empirically based but not empirically evaluated. This is different from Heikkila et al. (2015) study where papers with no empirical evaluation were not considered in the analysis. We also decided to categorize all the relevant studies identified based on SWEBOK (Bourque and Fairley, 2014) sub-topics. With this approach, it was possible to extract more information about requirements engineering in agile context.

4. Research method

In this research a systematic mapping study was developed following the guidelines proposed in Kitchenham and Charters (2007) and Petersen et al. (2008). In the planning activity it was possible to identify the need for the review, to specify the research questions, the research string and the inclusion/exclusion criteria.

4.1. Objectives and research questions

Agile software development approaches like XP (Beck and Andres, 2004), Scrum (Schwaber and Beedle, 2002) and lean thinking (Wang et al., 2012), have become more popular during the last few years and all of them share some common principles: improved customer satisfaction, adapting to change requirements, frequent delivering working software and collaboration between clients and developers. However, recent studies indicate high rates of projects failure, including those that are using agile processes. As previously described, according to the Standish Group (2014) five out of eight largest project cancellation factors are related to requirements engineering. Not much is known about requirements engineering in the context of agile software development. The community in general is trying to understand what exactly it is and how to deal with these difficulties. Motivated by this perception, the objective of this research is to get a deeper understanding on requirements engineering in the context of agile software development. For this it is intended to answer some specific questions as described below:

Table 2
Amount of papers selected during the first round of the systematic mapping study.

Databases	Total of articles selected	First round	
		Included	Excluded
ScienceDirect	300	10	290
SpringerLink	206	9	197
ACM	144	37	107
IEEExplore	1517	59	1458
Sub-Total	**2167**	**115**	**2052**
Snowballing	4	3	1
Total	**2171**	**118**	**2053**

1) On which requirements engineering topics are the researches on requirements engineering in agile software development concentrated?
2) What are the gaps concerning the requirements engineering in the context of agile software development?
3) What obstacles do the agile requirements engineering is facing (environment, people and resources)?

4.2. Search strategy

The study was conducted using four different databases: ACM, IEEExplore, ScienceDirect and SpringerLink. To define which study should be included, or not, inclusion and exclusion criteria were defined.

The inclusion criteria were:

1) Studies since 2001 until now. This date was defined because 2001 was the year of the Agile Manifesto release;
2) Studies in English language;
3) Studies related to the search string defined;
4) Peer reviewed studies;
5) Primary studies.

The exclusion criteria were:

1) The primary study is not labeled as paper published in journal or conference proceedings;
2) The subject was not directly related to agile requirements engineering;
3) Duplicated papers.

The search string defined was: *((software) AND (agile) AND ("requirements engineering" OR "requirement engineering"))*. The research was executed in February/March 2017 and returned 2167 results.

In order to identify additional relevant papers through the reference list found using the search string, a backward snowballing technique was used (Wohlin, 2014). This technique was previously employed in some recent researches (Jalali and Wohlin, 2012; Wohlin et al., 2013). By using this approach, four more papers were added. An extensive review of the selected studies was made by two researchers analyzing the title, the abstract and the conclusions, and applying the exclusion criteria.

A total of 118 papers were then selected as a result of this first round of classification while 2053 papers were excluded, as demonstrated in Table 2. To review the disagreements raised in this phase, some meetings were conducted to discuss and reach a consensus. When a consensus was not reached the complete paper was read and again the exclusion/inclusion criteria were applied.

Table 3
Criteria used for the quality assessment.

Minimum quality threshold of the review	1. The study reported is a research paper.
	2. The aims and objectives were clearly reported.
	3. There was an adequate description of the context in which the research was carried out.
Rigor	4. The research design was appropriate to address the aims of the research.
Credibility	5. The study provided clearly stated findings with credible results and justified conclusions.
Relevance	6. They provided value for research or practice.

Table 4
Amount of papers selected during the quality assessment.

Databases	Result of first round	Second round excluded	included
ScienceDirect	10	3	7
SpringerLink	9	0	9
ACM	37	8	29
IEEExplore	59	3	56
Snowballing	3	0	3
Total	**118**	**14**	**104**

4.2. Quality assessment

In this systematic mapping study a quality assessment was also used to filter the studies based on the criteria used by Dybå and Dingsøyr (2008). These quality assessment criteria provide a way to select the studies that are really satisfactory and that will contribute for the value of this research. The questions are related to the minimum quality threshold, rigor, credibility and relevance of the selected studies (Table 3). Criteria were graded on a dichotomous scale ("yes" or "no"). All studies that have at least one answer "no" on the first three questions were excluded because a minimum quality threshold was required for this review.

Once the quality assessment was applied, 14 studies were excluded as shown in Table 4. A total amount of 104 (shown in Appendix A) studies remained for the analysis and extraction of the results. Finally we applied an assessment to the remaining studies to find the inter-rater agreement between the two researchers that participated in the study search and selection. The value of *Kappa Coefficient*[1] was calculated and the result (0,662) was reached, which means good or substantial agreement.

4.3. Data extraction

For all of these studies, the following data were extracted: title, authors, country/location, database, publication source, year of publication, which requirements engineering areas the study belongs to, which type of evaluation was presented in the study, where the evaluation was applied (Industry or Academy), and Industry size.

4.4. Data classification

To answer the first research question it was necessary to perform some kind of classification of all selected studies. As the goal was to reach the "requirements engineering topics" the SWEBOK (Bourque and Fairley, 2014) was chosen as a referential because it is a broadly discussed and accepted document created under the sponsorship of the IEEE (Institute of Electrical and Electronics Engineers) (Sommerville, 2001) with the purpose of serving as a reference in subjects considered pertinent in the area of software engineering. The classification using the SWEBOK (Bourque and Fairley, 2014) sub-topics allowed us to have the view in extensions, not in depth, of the areas that agile requirements engineering can achieve. Through this classification it was possible to detect the areas with the highest concentration of research and also to score those where it is still possible a further exploration. So the classification also helped to emerge and clarify the gaps to answer the second and third research questions.

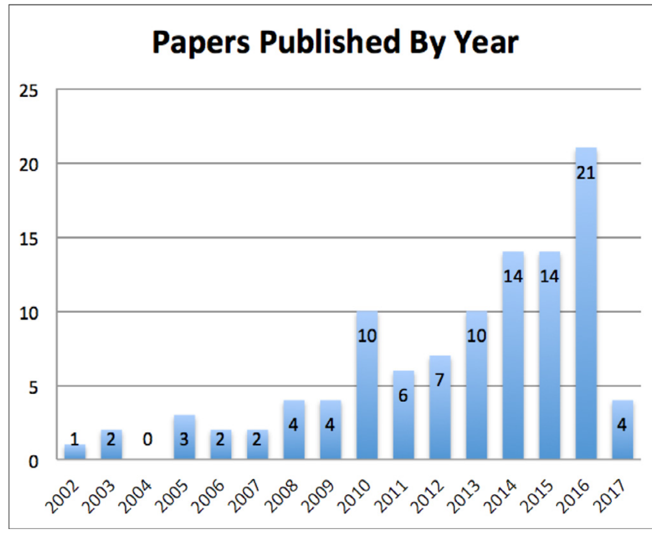

Fig. 3. Distribution of the published papers reviewed per year since the Agile Manifesto.

5. Results

In this section the results and the answers to the research questions are described.

5.1. Overview of the selected papers

The majority of the studies identified were published in conferences, journals and workshops as shown in Appendix B. Of the 104 identified studies, about 60 studies were published in conferences, 21 studies in journals, 18 studies in workshops, 2 studies in symposiums, 2 in a magazine, and 1 in a technical report. The results also suggest that some sources are preferred by researches. A great number of publication, 43 studies (41,34%) out of 104 were identified in: *IEEE International Requirements Engineering Conference* with 9 publications, *Requirements Engineering* with 7 publications, *IEEE SoutheastCon* and *Euromicro Conference on Software Engineering and Advanced Applications* both with 5 publications, *Information and Software Technology* and *International Conference on Research Challenges in Information Science* both with 4 publications, *Scientific Workshop XP, Empirical Software Engineering, International Working Conference on Requirements Engineering: Foundation for Software Quality* with 3 publications each. The remaining sources have only one or two researches published.

Regarding to the year of publication it is possible to suggest, analyzing Fig. 3, that this thematic research is increasing since the agile manifesto in 2001. The interest in agile software development is crescent but the community still does not have the whole knowledge and the completely understanding of what it is and how to deal with requirements engineering in the agile context. Although requirements engineering in agile software development has been studied, the overall understanding of RE in agile software development as a phenomenon is still weak (Heikkila et al., 2015).

[1] The Kappa Coefficient is a statistical measure of inter-rater agreement for qualitative items (Landis and Kosh, 1977).

Motivated by this need it is possible to see a tendency of more and more researchers getting deeper in understanding this approach. The low number of publications in 2017 can be explained because our search string was executed in the beginning of the year and this probably affected the result.

5.2. On which requirements engineering topics are the researches on requirements engineering in agile software development concentrated?

To answer the first research question a thematic classification was performed based on requirements engineering topics of SWEBOK (Bourque and Fairley, 2014). The aim of this classification was to get a global vision of the main topics explored by the researches in the context of agile requirements. To help us to complete this thematic classification we used the keywords mentioned in each research.

As shown in Fig. 4 the main subject explored by the researches was **4. Requirements Analysis- 4.4 Requirements Negotiation**.[2] It was found 15 researches about this thematic, which includes papers related to requirements prioritization ([S5], [S6], [S7], [S8], [S9], [S10], [S11], [S12], [S13], [S14]) and release planning ([S15], [S16], [S17], [S18], [S19]). It was possible to observe that, in 70% of researches related to requirements prioritization thematic two main researchers, Daneva and Racheva, were involved.

With 10 papers we observed three main topics. The first one is related to **1. Software Requirements Fundamentals - 1.3. Functional and Nonfunctional Requirements**. This thematic was found in [S20], [S21], [S22], [S23], [S24], [S25], [S26], [S27], [S28], [S29]. It is interesting to observe that 50% of the published papers about this thematic were published by the same researcher [S21], [S22], [S23], [S25], [S27]. So, probably W. M. Farid may be considered the main researcher about this thematic in the context of agile development.

The second one was related to **6. Requirements Validation – 6.4. Acceptance Test** ([S30], [S31], [S32], [S33], [S34], [S35], [S36], [S37], [S38], [S39]). It was not possible to identify one main researcher in the selected papers. But it was possible to observe that the weak alignment of RE with V&V is a problematic that may lead to problems in delivering the required products on time with the targeted quality. So many studies are bringing to light this problem and are presenting the solutions adopted.

The third one is related to **2. Requirements Process - 2.3. Process Support and Management.** About this thematic we found different approaches, which include papers related to requirements driven collaboration ([S40], [S41], [S42]), communication ([S43], [S44], [S45], [S46]), skills [S47], scope management [S48] and requirements management [S49].

With 9 papers we observed one main topic related to **2. Requirements Process - 2.3. Process Quality and improvement** ([S50], [S51], [S52], [S53], [S54], [S55], [S56], [S57], [S58]). Analyzing these papers it was possible to observe that 3 of them were published by Heck and Zaidman ([S53], [S55], [S57]), and 2 of them were published by G. Lucassen et al. ([S51], [S54]). They proposed different frameworks to work with quality criteria for agile requirements. User stories are widely accepted and used notation to document requirements in agile projects but despite their popularity little attention is given to assess their quality. Existing approaches to user story quality employ highly qualitative metrics, such as the heuristics of the INVEST (Independent-Negotiable-Valuable-Estimable-Scalable-Testable) (Wake, 2003), but user stories are too often poorly written in practice and exhibit inherent quality defects [S54].

The remaining papers were all published by different authors. In [S50] it is showed how a method for elicitation, documentation and validation of software user requirements (MEDoV) supports agile and lean software development methodologies. The developed method uses recommendations of ISO standards to improve the quality of software. In [S52] an investigation through a case study showed the relationship between the defects in requirements and the impact on the accuracy of the resulting functional size, project relative effort and measurement. Kalinowski et al. [S56] proposed empirically based guidelines that can be used by different types of organizations according to their size (small, medium or large) and process model (agile or plan-driven) to help them in preventing problems related to requirements engineering. Gebhart et al. proposed in [S58] an enhancement of scenario-based requirements engineering techniques, for the development of RESTful web services, that results in requirements that fulfill the quality characteristics for software requirements specifications of the international standard ISO/IEC/IEEE 29148 (ISO/IEC/IEEE, ISO/IEC/IEEE 2011).

With 7 papers we observed three main topics. The first one was related to **5. Requirement Specifications.** Analyzing this area no main researcher was found in the selected papers ([S59], [S60], [S61], [S62], [S63], [S64], [S65]). Different authors wrote all of them. They discuss a big challenge on agile software development that is the lack of documentation and how to deal with it. Rubin and Rubin [S59] developed an approach that enables incorporating domain documentation to agile development, while keeping the process adaptive through active documentation. Faegri and Moe [S60] try to understand the requirements engineering process in agile context applied a model of software development as a conversation, where verbal and direct communication were encouraged while written and indirect communication were discouraged. Nawracki et al. [S61] proposed how to introduce documented requirements to XP. Valencia, Oliveira and Sim [S62] show the results of a controlled experiment aimed to learn whether use cases could help agile requirements, and indirectly, to find if agile requirements techniques are sufficient. Stettina and Kroom [S63] explored it in a study with 30 teams in a small to large size projects, looking at the actions the teams take and the artifacts they use during a software project handover. Lopez-Nores et al. [S64] introduced an agile approach, with low computational cost to achieve frequent interaction with the stakeholders, which helps identify suitable evolutions of a specification. Lucassen et al. [S65] explored how practitioners perceive the usage of user stories to capture requirements in agile software development.

The second topic, with 7 papers, was related to **3. Requirement Elicitation – 3.2 Elicitation Technique** ([S66], [S67], [S68], [S69], [S70], [S71], [S72]). Different techniques were explored in these papers including aspect-oriented approach [S66], JAD [S67], prototyping [S68], mind-map [S69], query-based requirements engineering [S70], simulations [S71] and gamification [S72]. It is interesting that some of their findings indicates that the use of these techniques can help with some challenges of agile RE as lack of documentation, motivation for RE and poor quality communication.

The third topic, with 7 papers, was related to **7. Practical Considerations - 7.4 Requirements Tracing.** ([S73], [S74], [S75], [S76], [S77], [S78], [S79]). Different approaches were discussed in the papers including traceability ([S73], [S76], [S77], [S78], [S79]) and also requirements dependencies ([S74], [S75]). Traceability relations can support several software development activities such as evolution, reuse, validation, rationale, understanding, and change impact analysis [S79]. Despite of finding a reasonable number of studies, it was possible to detect that there are a big number of requirements traceability models and tools available without a roadmap to provide a guidance to evaluate them. This is probably caused because on agile methodologies there is a tendency on

[2] We are using the same labeling scheme as SWEBOK.

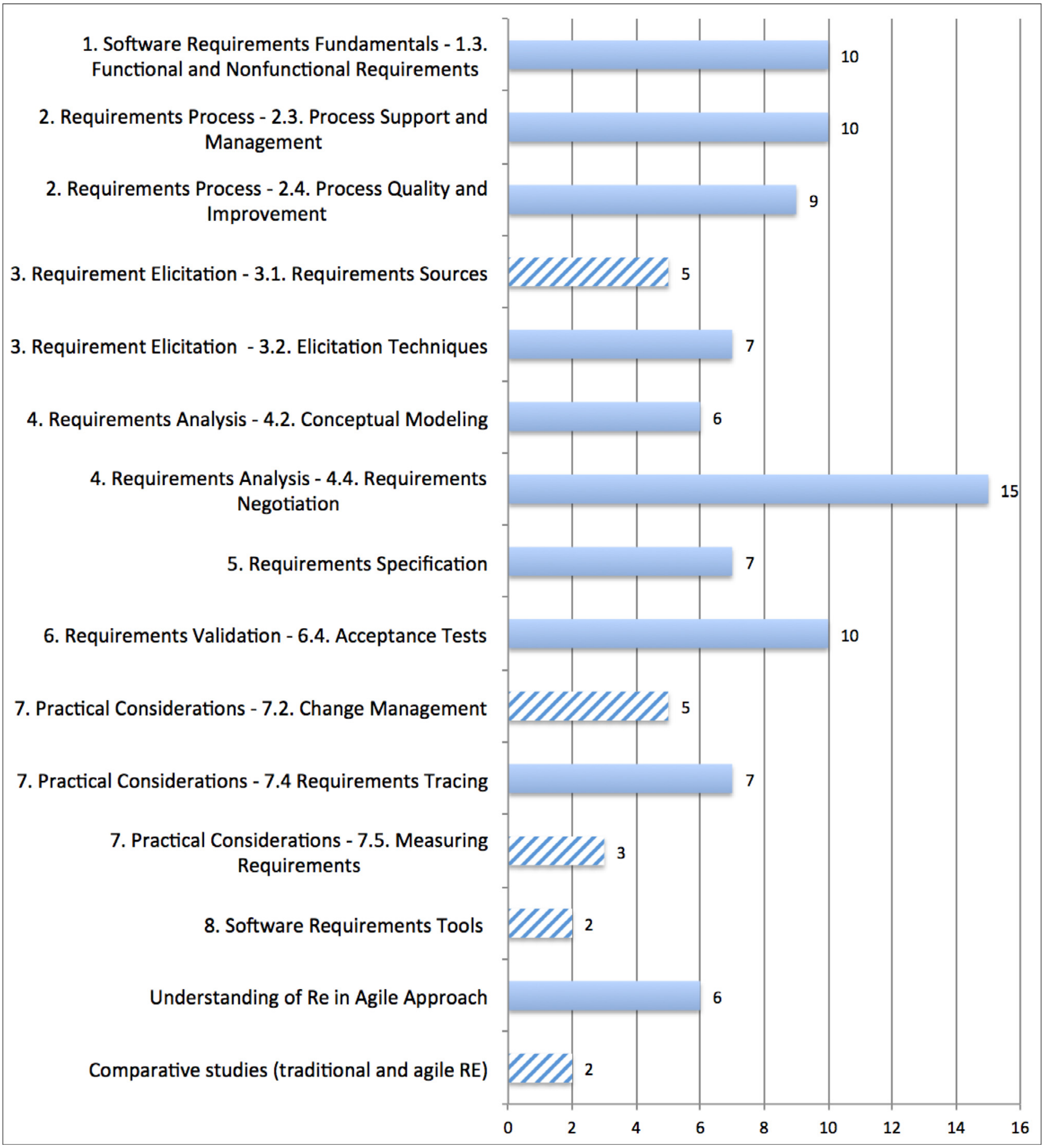

Fig. 4. Distribution of published reviewed researches using a thematic classification.

deliver business value as the main prioritization criteria. Documentation and traceability sometimes are seen as a heavy process by agile developers, because it is hard to create and maintain, so they are usually neglected or run in background. Unmanaged dependencies among requirements have been attributed to stakeholder's dissatisfaction and numerous software failures. So both can directly impact on the cost of the software.

With 6 papers we observed two main topics. The first one was related to **4. Requirements Analysis - 4.2. Conceptual Modeling** ([S80], [S81], [S82], [S83], [S84], [S85]). Analyzing the papers it was possible to detect that inside the agile software development the improvement of the communication process by continuous stakeholders' involvement is necessary. So through the requirements models is possible to improve the transparency and to reduce the cognitive load.

The second one was related to **Understanding the Requirements engineering in Agile Approach** As it was difficult to classify some of the papers using the SWEBOK (Bourque and Fairley, 2014) structure, in the present study it was decided to create a different sub-area and aggregate them to facilitate the visualization of the results. Analyzing the six papers found for Understanding the Requirements engineering in Agile Approach ([S4], [S86], [S87], [S88], [S89], [S1]) it was not possible to identify only one main researcher. It was possible to conclude that different researchers are

Table 5
Evaluation type.

Evaluation type	Amount of studies	Related studies
Empirical	50	[S7], [S14], [S20], [S21], [S22], [S23], [S24], [S25], [S27], [S29], [S34], [S36], [S37], [S38], [S39], [S40], [S46], [S48], [S49], [S50], [S51], [S53], [S54], [S55], [S57], [S58], [S59], [S61], [S62], [S64], [S65], [S69], [S72], [S76], [S77], [S78], [S79], [S80], [S81], [S82], [S83], [S85], [S90], [S91], S92], [S93], [S94], [S95], [S96], [S97]
Not empirical	25	[S3], [S5], [S6], [S9], [S10], [S11], [S16], [S17], [S28], [S32], [S42], [S45], [S52], [S56], [S60], [S68], [S71], [S73], [S74], [S86], [S89], [S98], [S99], [S100], [S101]
Not applied	29	[S1], [S2], [S4], [S8], [S12], [S13], [S15], [S18], [S19], [S26], [S30], [S31], [S33], [S35], [S41], [S43], [S44], [S47], [S63], [S66], [S67], [S70], [S75], [S84], [S87], [S88], [S102], [S103], [S104]

Table 6
Identification of industry size.

Industry size	Amount of studies	Related studies
Startups	–	
Small	2	[S91], [S93]
Medium	2	[S46], [S72]
Big	9	[S29], [S40], [S48], [S49], [S78], [S79], [S81], [S85], [S90]
Not informed	11	[S14], [S34], [S36], [S50], [S51], [S54], [S62], [S65], [S77], [S80], [S95]

getting involved to this thematic. This reinforces the necessity of the community on understanding the role of requirements engineering in agile context.

5.3. What are the gaps concerning the requirements engineering in the context of agile software development?

During the analysis of the papers it was possible to identify two kinds of gaps: generic and specific. When some information extracted from the papers was grouped, it was possible to identify some general gaps. The other specific gaps where extracted from individual analysis of each paper.

5.3.1. General gaps

1) **Lack of empirical evaluation studies**: In this research another classification was made to identify the studies that were empirically evaluated. According to the classification presented by Wohlin et al. (2003) studies that evaluated their proposal using controlled experiments, case studies, surveys or a post-mortem analysis can be considered empirically evaluated. Studies that used other tools (simulations, grounded theory, comparative studies, measurements or focus group) to evaluate their results were not considered empirically evaluated. In this analysis it was possible to identify 50 studies that have performed an empirical evaluation, 25 studies have performed a not empirical evaluation and 29 studies that had not presented an evaluation as shown on Table 5. The numbers confirm that despite a great number of empirical studies, more than 50% are represented by studies with no empirical validation or with no validation. In this research we decided to include not empirical and not applied evaluation studies based in a different view proposed by Wohlin et al. (2003), which reinforces that a category of solutions proposals would contain papers with no empirical evidences.

 1.1) **Lack of studies exploring startups:** During the classification of the selected papers it was possible to detect that 24 out of 50 were studies empirically evaluated in Industry, 23 studies were empirically evaluated in Academy and 3 were evaluated in Academy/Industry. All of 24 papers empirically evaluated in Industry were classified with the aim of identifying the size of the Industry, as shown on Table 6. Some of the sources did not inform the size of the industry where the evaluation occurred or any other information that could provide aid for the classification. No papers were found exploring the context of startups or medium sized industry. It is an interesting information because during the last few years many startups and spin-offs have been emerged and it was expected to find researches in this area. Startups are dynamics and the use of agile development could fit their needs.

2) **Neglecting quality requirements**

Despite of the great number of researches related to non-functional requirements (10) as shown graphically in Fig. 4, during the analysis it was possible to detect the need of exploring other topics related to quality requirements. There is a consensus in the research community that agile software development methodologies are becoming more and more popular. But it is also clear that agile methodologies have not adequately modeled non-functional requirements and their potential solutions [S21]. Agile development methods are sometimes criticized for not having explicit practices for non-functional requirements. They can determine some aspects of the new system like usability, security, portability, maintainability and performance. It is possible to see some researches trying to solve this problematic as presented in [S21] with NORMATIC. This is a Java-based simulation tool for modeling non-functional requirements for semi-automatic processes. NORMAP (a lightweight engineering of NFRs for agile processes) is also another solution presented in [S22]. In [S23] a project management and requirements quality metrics that would be used to design a risk-driven algorithm to prioritize and plan an improved requirements implementation is presented (NORPLAN). Aspects related to maintainability, portability, security or performance are also often neglected probably because of the tendency in the use of business value as the main prioritization criteria [S68].

This negative tendency of neglecting quality requirements probably emerged due to the fact that in agile methods the use of minimal documentation is intrinsic [S61]. This brings to light the erroneous impression that quality is not necessary or that it can be treated in background.

5.3.2. Specific gaps

This topic presents the result of another analysis applied over the selected studies. At first all studies were classified under a specific SWEBOK (Bourque and Fairley, 2014) sub-topic (thematic classification). This classification helped to separate the articles in order to analyze the proposed solutions as the aim was to emerge specific gaps to provide aids for further research. As represented in Fig. 4**,** all sub-topics until 1/3 of the biggest value (15) of the total amount of papers found were considered specific gaps and were separately identified.

The first sub-topic that was considered as being a specific gap was related to **3. Requirement Elicitation - 3.1. Requirements Sources**. This SWEBOK sub-topic is designed to promote awareness of the various sources of software requirements and of the frameworks for managing them (goals, domain knowledge, stakeholders, business rules, operational environment, organizational environment).

In this sub-area only five papers were found. The first paper makes reference to “elicitation of goals“ [S92]. In this paper an Agile Technique for Agent Based Goal Elicitation (ATABGE), based on the mechanisms of agile practices and approaches for extracting the goals from stakeholders is presented. The second study found [S93], presented a Knowledge assisted Agile Requirements Evolution (K-agileRE). It emphasizes the fact that providing a domain knowledge edge can impart agility to requirements definition. The third study [S99] is about experiences with user-center design (UCD) and agile requirements engineering (RE) in fixed price projects. The forth study explored the agile requirements elicitation approach based on NFR and business process models [S91]. The aim of this paper is to demonstrate how several models and techniques such as goals, business process models, patterns and non-functional requirements have helped in defining software requirements of the micro-business. The fifth study [S96] presented a business process ontology for defining user story. This process is an alternative to write user stories more easily, using the collected knowledge concepts of related or similar business process, which have been previous successfully done. We observed after analyzing the papers that each one focused on a specific source. It provides space to explore them deeper.

The second sub-topic that was considered a specific gap was related to **7. Practical Considerations - 7.2. Change Management.** This sub-topic is central to the management of requirements. In this sub-area only five papers were found ([S90], [S94], [S95], [S98], [S103]). Two papers investigate how to deal with the rapid change in requirements when working with agile software development and how to accommodate them [S94], [S103]. Some of them also investigate the commonalities and differences between agile and traditional approaches (document-driven, V-model) in managing uncertainty in requirement gathering [S95], [S98]. As requirements changes are very expensive to accommodate, this topic became an interesting source of research. In [S90] the authors presented a framework, called RE-KOMBINE, which examine how to support lightweight agile requirements process, which can be systematically modeled, analyzed and changed.

The third sub-topic that was considered a specific gap was related to **7. Practical Considerations - 7.5. Measuring Requirements.** This sub-topic is related to the concept of “volume“ which is useful in evaluating the “size” of a change in requirements, in estimating the cost of a development or maintenance task, or simply for using as the denominator in other measurement. In this sub-area only three papers were found ([S97], [S100], [S101]). In the first paper [S100] it is presented a Bayesian network model suitable for effort prediction in agile methods. The main objective of this research is to find a technique that will facilitate the assessment of the required effort. The second paper [S101] developed a replication study based on a method proposed by Hussain, Kosseim and Ormandjieva (HKO) that can be used to automatically classify textual requirements with respect to their COSMIC functional size. In this paper it was considered how to support agile teams by providing them additional information that might help them to estimate effort more accurately. The third paper [S97] reported the results of a study that examines the effect of using risk and effort annotations on project teams' level of shared understanding, estimation accuracy and estimation bias. The results showed that the use of annotations in planning poker increases estimation accuracy.

Agile development teams usually use story points/planning poker as a technique to measure the effort needed to implement a user story. Both are expert estimates techniques that are based on stakeholders' knowledge. We observed that the accuracy of effort estimates is an important factor for the success of a project but unfortunately agile teams often underestimate development effort.

The fourth sub-topic that was considered a specific gap was related to **8. Software Requirements Tools.** This SWEBOK sub-topic is related to tools for dealing with software requirements and it fall broadly into two categories: tools for modeling and tools for managing requirements. In this sub-area only two papers were found [S102], [S104]. In the first paper [S102], developed a comparative study of software tools for user story management. In the second paper [S104] is provided an overview of the central conceptual ideas for an agile documentation tool.

A great number of software tools for agile project management has emerged in recent years, providing support to practices based on user stories. However, each one offers different functionalities and a more detailed review of them is needed. So more researches could be developed to acquire a better insight into this topic, exploring large number of practitioners in various roles with different backgrounds. Another thing that was possible to detect during the analysis is that no researches were found exploring and comparing tools for modeling requirements.

The last specific gap identified is related to **Comparative Studies**. Related to this topic only two researches were found. The first one [S2] analyzes commonalities and differences of both approaches and determines possible ways of how agile software development can benefit from requirements engineering methods. In the second research [S3] the authors distinguishes the traditional RE and Agile RE and investigates the reasons for which software industries shifted from traditional RE to Agile RE. More studies could be developed comparing agile RE and traditional RE to present more discussion about the strengths and weaknesses of each approach. No researches were found to present quantitative results to base the arguments. [S3] is an empirically based study but the evaluation was through experts' interviews.

5.4. What obstacles do the agile requirements engineering is facing (environment, people and resources)?

In this research it was possible to detect a list of different obstacles of using agile RE as described below.

1) **Environment**
   1.1 **Difficulties with communication in distributed teams** are a big challenge in agile projects. Projects that involve a large number of partners, target groups of users and developers are more apparent [S7]. In [S88] the author get deeper in understanding what are the main challenges related to agile requirements and what are the important themes for research on agile requirements in large-scale. Communication and coordination appeared as being highly relevant including in the context of large-scale. In [S41] the authors explored the Requirement Driven Collaboration and the sociotechnical aspects involved in the process of requirements engineering. This is crucial and utmost importance in case of agile methods, especially for geographically distributed teams.
2) **People**
   2.1 **Difficulties on finding specific and specialized skills** (Heikkila et al., 2015). Agile RE requires highly skilled people and personnel turnover is a trouble since most requirements knowledge is tacit [S86]. In [S99] for example it is presented an experience with user-centered design (UCD) and agile requirements engineering in a fixed-price project. UCD allows RE to focus on users experiences (UX) and in this project they introduced a new role in the development team called “On-site User Experience Consultant”. Sometimes in an attempt to solve one challenge of RE agile, for example, the motivation issues related to requirements work, the solution adopted was to use acceptance test driven. However,

the adoption of TDD is often undermined by lack of skills and motivation to learn [S86].

2.2 **Motivation issues related to RE work**. To keep the documentation up to date, despite of being minimal, can be a daunting task for many teams [S68]. The utilization of automated acceptance test cases was believed to be more motivating [S87].

2.3 **Customer availability**. On agile software development it is not necessary but crucial for the success of the project. The intense interaction between the developers and the costumers is hard because it involves some variables like time, budget allocation and domain knowledge (Inayat et al., 2015), [S4], (Heikkila et al., 2015).

2.4 **Customer inability and agreement** refers to the lack of knowledge of customer to define the requirements and his inability in terms of decision making (Inayat et al., 2015). This is also described in [S5] when a group of customer is involved. This can direct affect the performance of the project.

2.5 **Lack of accuracy on estimations**. Some techniques like use story points/planning poker are used by agile development teams usually to measure the effort needed to implement a user story. Both techniques are based on stakeholders' knowledge. Of course that expert estimates are particularly useful if there is a lack of historical data on previous projects or a company's estimation resources are limited, but it can bring others problems in case of no prior experience [S101]. Planning poker seems to encourage extreme estimates, which could be traced to group polarization.

3) **Resources**

3.1 **Lack of specialized tool** is also another weakness as shown by the qualitative analysis in [S102]. It shows that agile professionals invest considerable time and effort to find a tool that support their processes and suit their need in general. The conclusion of this qualitative analysis of users review demonstrate that practitioners prefer tools that are easy to set up, easy to learn, easy to use, and easy to customize, over more sophisticated tools.

3.2 **Lack of documentation** is considered one of the biggest weaknesses of agile RE. In Cao and Ramesh (2008), for example, is described this problematic through the insufficiency of the user story formats. Consistency and verifiability of user stories are difficult to validate. An agile software development team relies on communication and collaboration to perform requirements engineering activities, rather than dedicated analysis tools or documentation [S44]. Users of XP (Beck and Andres, 2004) (that relies only on code and test cases) also recognized this problematic and from requirements engineering point of view the most important issues are: maintenance and lack of wider perspective of the system to be build [S50].

3.3 **Inappropriate architecture** sometimes is a consequence of the decisions taken by the team in the early stages of the project [S86]. This only appears as a real problem when new requirement are introduced to the project. Developers make use of refactoring to solve this problem but it can add costs to the project (Inayat et al., 2015).

3.4 **Growing technical debt** can be a consequence of the use of some agile RE practices. It is described in Heikkila et al. (2015) and may lead to inadequate architecture due to the rapidly responses to attend the clients expectations.

3.5 **Imprecise cost and schedule estimation** is presented as a weakness in some researches. The first one to explore this fragility was [S86], where some challenges of agile requirements engineering were exposed. One of them is the initial cost and schedule estimation that are completely based on user stories. In [S52] the requirements engineering quality is revealed through functional size measurement. They argue that defects in the requirements phase may have impact on the accuracy of functional size and can impact on the project effort.

6. Discussion

As already mentioned the goal of this study is to explore the agile requirements engineering and bring to light more discussions promoted by literature foundations. During this research we selected 104 papers related to agile requirements engineering. To attend this goal some research questions were formulated and answered. In this section some important aspects that emerged in this research are going to be discussed.

On Section 3 some related studies like systematic literature reviews and mapping studies were presented. As previous mentioned Heikkila et al. (2015) had already made a systematic mapping study in the agile requirement engineering but it was not possible to identify in their work what are the less explored or not completed explored topics regarding to requirements engineering in agile context. So as it was not clear what are the gaps related to requirements engineering in agile context we decided to propose a different approach, based on SWEBOK sub-topics, to make it explicit. After completing our analysis it was possible to detect some gaps, which can be further explored like **Requirement Elicitation - Requirements Sources, Change Management, Measuring Requirement, Software Requirements Tools and Comparative Studies.**

Related to specific gaps identified one of them need special attention. As previous mentioned we make the inclusion of studies with no empirical evaluation. We did not follow the recommendation for developing systematic studies that is to discard them. They were included because a different view proposed by Wieringa et al. (2006), which reinforces that a category of solutions proposals would contain papers with no empirical evidences, was used. Another point is that we did not have a parameter to compare the results and affirm that there is a lack of empirically evaluated study.

This point was also previous detected by some authors described on related studies (Inayat et al., 2015; Heikkila et al., 2015). Now with the classification presented in this papers using empirically evaluated, not applied or not empirically evaluated, we can have a parameter of comparison. As the results showed that more than 50% of the papers were not evaluated or were not empirically evaluated it is possible to conclude that the researchers need to be encouraged to empirically validate their researches and demonstrate the results to the community.

Furthermore, the results showed a lack of studies exploring startups. As recently many startups and spin-offs have been emerged, it was expected to find more researches in this area. Startups are dynamics and the use of agile development fits on their reality. It was expected to find a better distribution of data, exploring mainly startups, small and medium organizations.

Another gap that appeared in Section 5.3 is the neglecting quality requirements. Despite of its popularity agile methodologies have not adequately modeled non-functional requirements and their potential solutions [S21]. This negative tendency of neglecting quality requirements probably emerged because the fact that in agile methodologies the use of minimal documentation is intrinsic [S61]. This brings to light the erroneous impression that quality is not necessary or that can be treated in background.

It is important to highlight that "quality requirements" is different of "quality of requirements" . We observed that some recent researches are working with this last thematic [S15], [S51], [S53],

[S55], [S57], [S58] and that is why it appears on Fig. 3 as a relevant area of research **2. Requirements Process - 2.3. Process Quality and Improvement.**

In Section 5.4 it was described some of the obstacles the practitioners might have to face when using requirements engineering in agile context. The obstacles were classified in three different categories: environment, people and resources. This topic is interesting because many organizations are becoming attracted by the promise of a faster time to market through agility, but they always have to keep in mind the context and take actions to mitigate the challenges incurred. It brings new challenges for the whole organization and especially for requirements engineering.

7. Threats to validity

The first threat to the validity of this study is related to the bias on the process of selection of the studies. To reduce this bias the guidelines proposed in Kitchenham and Charters (2007) and Petersen et al. (2008) were used. The first difficulty appeared on preparing the search string. The first string that was proposed by the authors, with more than 15 attributes related to agile requirements engineering, could not be used in all databases. Some of them have restrictions related to the number of attributes used, so it was not possible to implement it. Because of this, the search string was reevaluated and it was focused on the main keywords considered relevant to the research questions to make it as lean as possible.

After having the research string applied and to reduce the bias in the process of selecting the relevant studies it was decided to examine the selected papers in pairs. In this case two researchers evaluated the selected studies in a peer reviewed manner. All replicated studies were extracted and at the end of this process an inter-rate agreement coefficient was calculated. The *kappa coefficient* returned was 0,662 which means that a substantial agreement was reached.

Another point that can be a threat to the validity of the study is related to the number of selected databases. This systematic mapping study was developed using four databases (SpringerLink, ScienceDirect, ACM and IEEExplorer) because they were considered the most important and relevant databases for the goal of this research. However, additional databases could have produced complementary missed information. We also decided not to include a manual search on this work because this step probably could not be replicated by the other researches, so we considered it out of our scope.

We considered another threat to the validity the inclusion of studies with no empirical evaluation as the recommendation for developing systematic studies is to discard them (Kitchenham and Charters, 2007). They were included because a different view proposed by Wieringa et al. (2006), which reinforces that a category of solutions proposals would contain papers with no empirical evidences, was used.

Besides these issues already mentioned, the fact that only one of the authors have made the assessment of the quality of papers can also be considered a vulnerability factor. However, as only few studies that did not have the minimum quality were excluded, this was not considered a risk factor for the research. On the other hand it was considered a quality improvement criteria.

8. Conclusion

This paper presented the results of a systematic mapping study on requirements engineering in agile software development context. In this review a total of 104 studies, published between 2001 and March of 2017, were selected to be reviewed and analyzed. The goal of this research was to map the subject area of requirements engineering in agile context to identify the main topics that have been studied related to requirements engineering in agile context and also to identify gaps in the literature to develop future researches. It was also intended to identify the obstacles related to environment, people or resources that the practitioners might have to face when using requirements engineering in agile context.

It was possible to identify 15 areas (13 based on SWEBOK (Bourque and Fairley, 2014) sub-topics) where the researches were developed. The concentration of the researches remained in 10 main areas. The first topic more discussed was Requirements Analysis - Requirements Negotiation with 15 papers found. With 10 papers we observed three main topics. The first one is related to 1. Software Requirements Fundamentals - 1.3. Functional and Nonfunctional Requirements, the second one was related to 6. Requirements Validation – 6.4. Acceptance and the third one is related to 2. Requirements Process - 2.3. Process Support and Management. With 9 papers we observed one main topic related to 2. Requirements Process - 2.3. Process Quality and Improvement. With 7 papers we observed three main topics. The first one was related to 5. Requirement Specifications, the second topic, with 7 papers, was related to 3. Requirement Elicitation – 3.2 Elicitation Technique, the third topic, with 7 papers, was related to 7. Practical Considerations - 7.4 Requirements Tracing. With 6 papers we observed two main topics. The first one was related to 4. Requirements Analysis - 4.2. Conceptual Modeling. The second one was related to Understanding the Requirements engineering in Agile Approach. As it was difficult to classify some of the papers using the SWEBOK (Bourque and Fairley, 2014) structure, in the present study it was decided to create a different sub-area and aggregate them to facilitate the visualization of the results but that together can represent an area to be explored.

As in the other five remaining areas the number of studies were limited (with five or less papers), they were used to explore gaps for future researches. We separated the gaps in two categories: general gaps and specific gaps. In the general gaps it was possible to identify a lack of empirically evaluated studies, lack of studies related to startups where the studies were developed and the need of exploring other topics related to quality requirements. In relation to specific gaps topics like requirement elicitation - requirements sources, change management, measuring requirements, software requirements tools and comparative studies, need special attention.

Another goal of this research was to identify some obstacles that the practitioners might have to face dealing with requirements engineering in agile context. During the analysis it was possible to identify one obstacle related to environment (difficulties with communication in distributed teams), five obstacles related to people (difficulties on finding specific and specialized skills, motivation issues, customer availability, customer inability and agreement, lack of accuracy on estimations) and five obstacles related to resources (lack of specialized too, lack of documentation, inappropriate architecture, growing technical debt, imprecise cost and schedule estimation).

The conclusion of this work is that the field of requirements engineering in agile context is broad and we could demonstrate it through the classification of the papers in 13 SWEBOK subtopics. We also showed that despite of the obstacles demonstrated in Section 5.4, the interest in this thematic research is increasing since the agile manifesto in 2001**.** Furthermore it is noticed that requirement engineering in agile software development is still immature and need more researches, mainly including empirical evaluation studies. It is expected that this work will also serve as an initial source for future work.

Appendix A. List of studies selected for the review (primary sources)

[S1] De Lucia, A., Qusef, A., 2010. Requirements engineering in agile software development. J. Emerg. Technol. Web Intell. 2, 212–220. doi:10.4304/jetwi.2.3.212-220.

[S2] Paetsch, F., Eberlein, A., Maurer, F., 2003. Requirements engineering and agile software development. WET ICE 2003. Proceedings. Twelfth IEEE Int. Work. Enabling Technol. Infrastruct. Collab. Enterp. 2003. 308–313. doi:10.1109/ENABL.2003.1231428.

[S3] Batool, a, Motla, Y.H., Hamid, B., Asghar, S., Riaz, M., Mukhtar, M., Ahmed, M., 2013. Comparative study of traditional requirement engineering and Agile requirement engineering. Adv. Commun. Technol. (ICACT), 2013 15th Int. Conf. 1006–1014.

[S4] Inayat, I., Moraes, L., Daneva, M., Salim, S.S., 2015. A Reflection on Agile Requirements Engineering: Solutions Brought and Challenges Posed. Sci. Work. Proc. XP2015 6:1–6:7. doi:10.1145/2764979.2764985.

[S5] Daneva, M., van der Veen, E., Amrit, C., Ghaisas, S., Sikkel, K., Kumar, R., Ajmeri, N., Ramteerthkar, U., Wieringa, R., 2013. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. J. Syst. Softw. 86, 1333–1353. doi:10.1016/j.jss.2012.12.046.

[S6] Port, D., Olkov, A., Menzies, T., 2008. Using simulation to investigate requirements prioritization strategies. ASE 2008 - 23rd IEEE/ACM Int. Conf. Autom. Softw. Eng. Proc. 268–277. doi:10.1109/ASE.2008.37.

[S7] Belsis, P., Koutoumanos, A., Sgouropoulou, C., 2014. PBURC: A patterns-based, unsupervised requirements clustering framework for distributed agile software development. Requir. Eng. 19, 213–225. doi:10.1007/s00766-013-0172-9.

[S8] Racheva, Z., Daneva, M., 2010. How do real options concepts fit in agile requirements engineering? 8th ACIS Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2010 231–238. doi:10.1109/SERA.2010.37.

[S9] Racheva, Z., Daneva, M., Sikkel, K., Wieringa, R., Herrmann, A., 2010. Do we know enough about requirements prioritization in agile projects: Insights from a case study. Proc. 2010 18th IEEE Int. Requir. Eng. Conf. RE2010 147–156. doi:10.1109/RE.2010.27.

[S10] Herrmann, A., Bakalova, Z., Daneva, M., Herrmann, A., Wieringa, R., 2016. Agile Requirements Prioritization : What Happens in Practice and What Is Described in Literature Agile Requirements Prioritization : What Happens in Practice and What Is Described in Literature 181–195. doi:10.1007/978-3-642-19858-8.

[S11] Racheva, Z., Daneva, M., Herrmann, A., 2010. A Conceptual Model of Client-driven Agile Requirements Prioritization: Results of a Case Study. Proc. 2010 {ACM-IEEE} Int. Symp. Empir. Softw. Eng. Meas. 39:1–39:4. doi:10.1145/1852786.1852837.

[S12] Racheva, Z., Daneva, M., Herrmann, A., Wieringa, R.J., 2010. A Conceptual Model and Process for Client-driven Agile Requirements Prioritization. 2010 Fourth Int. Conf. Res. Challenges Inf. Sci. 287–298. doi:10.1109/RCIS.2010.5507388.

[S13] Racheva, Z., Daneva, M., Buglione, L., 2008. Supporting the dynamic reprioritization of requirements in agile development of software products. 2008 2nd Int. Work. Softw. Prod. Manag. ISWPM'08. doi:10.1109/IWSPM.2008.7.

[S14] Popli, R., Chauhan, N., Sharma, H., 2014. Prioritising user stories in agile environment. 2014 Int. Conf. Issues Challenges Intell. Comput. Tech. 515–519. doi:10.1109/ICICICT.2014.6781336.

[S15] Heikkilä, V.T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T., Järvinen, J., 2015. Operational release planning in large-scale Scrum with multiple stakeholders – A longitudinal case study at F-Secure Corporation. Inf. Softw. Technol. 57, 116–140. doi:10.1016/j.infsof.2014.09.005.

[S16] Li, C., van den Akker, M., Brinkkemper, S., Diepen, G., 2010. An integrated approach for requirement selection and scheduling in software release planning. Requir. Eng. 15, 375–396. doi:10.1007/s00766-010-0104-x.

[S17] del Sagrado, J., del Águila, I.M., Orellana, F.J., 2015. Multi-objective ant colony optimization for requirements selection. Empir. Softw. Eng. 20, 577–610. doi:10.1007/s10664-013-9287-3.

[S18] Heikkilä, V., Rautiainen, K., Jansen, S., 2010. A revelatory case study on scaling agile release planning. Proc. - 36th EUROMICRO Conf. Softw. Eng. Adv. Appl. SEAA 2010 289–296. doi:10.1109/SEAA.2010.37.

[S19] Gillain, J., Jureta, I., & Faulkner, S. (2016, September). Planning Optimal Agile Releases via Requirements Optimization. In Requirements Engineering Conference Workshops (REW), IEEE International (pp. 10–16). IEEE. doi:10.1109/REW.2016.36.

[S20] Bourimi, M., Kesdogan, D., 2013. Experiences by using AFFINE for building collaborative applications for online communities. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 8029 LNCS, 345–354. doi:10.1007/978-3-642-39371-6-39.

[S21] Farid, W.M., Mitropoulos, F.J., 2012. NORMATIC: A visual tool for modeling non-functional requirements in agile processes. Conf. Proc. - IEEE SOUTHEASTCON. doi:10.1109/SECon.2012.6196989.

[S22] Farid, W.M., 2012. The Normap methodology: Lightweight engineering of non-functional requirements for agile processes. Proc. - Asia-Pacific Softw. Eng. Conf. APSEC 1, 322–325. doi:10.1109/APSEC.2012.23.

[S23] Farid, W.M., Mitropoulos, F.J., 2013. NORPLAN: Non-functional requirements planning for agile processes. Conf. Proc. - IEEE SOUTHEASTCON. doi:10.1109/SECON.2013.6567463.

[S24] Domah, D., Mitropoulos, F.J., 2015. The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development. SoutheastCon 2015 1–7. doi:10.1109/SECON.2015.7133028.

[S25] Farid, W.M., Mitropoulos, F.J., 2013. Visualization and scheduling of non-functional requirements for agile processes. Conf. Proc. - IEEE SOUTHEASTCON. doi:10.1109/SECON.2013.6567413.

[S26] Binti Arbain, A.F., Ghani, I., Wan Kadir, W.M.N., 2014. Agile non functional requiremnents (NFR) traceability metamodel. 2014 8th Malaysian Softw. Eng. Conf. MySEC 2014 228–233. doi:10.1109/MySec.2014.6986019.

[S27] Farid, W.M., Mitropoulos, F.J., 2012. Novel lightweight engineering artifacts for modeling non-functional requirements in agile processes. Conf. Proc. - IEEE SOUTHEASTCON. doi:10.1109/SECon.2012.6196988.

[S28] Aljallabi, B.M., 2015. Enhancement Approach for Non-Functional Requirements Analysis in Agile Environment. doi:10.1109/ICCNEEE.2015.7381407.

[S29] Ho, C.W., Johnson, M.J., Williams, L., Maximilien, E.M., 2006. On Agile performance requirements specification and testing. Proc. - Agil. Conf. 2006 2006, 47–52. doi:10.1109/AGILE.2006.41.

[S30] Bjarnason, E., Runeson, P., Borg, M., Unterkalmsteiner, M., Engström, E., Regnell, B., Sabaliauskaite, G., Loconsole, A., Gorschek, T., Feldt, R., 2014. Challenges and practices in aligning requirements with verification and validation: a case study of six companies. Empir. Softw. Eng. 19, 1809–1855. doi:10.1007/s10664-013-9263-y.

[S31] Haugset, B., Stålhane, T., 2011. Automated acceptance testing as an agile requirements engineering practice. Proc. Annu. Hawaii Int. Conf. Syst. Sci. 5289–5298. doi:10.1109/HICSS.2012.127.

[S32] Gallardo-Valencia, R.E., Sim, S.E., 2009. Continuous and collaborative validation: A field study of requirements knowledge in agile. 2009 2nd Int. Work. Manag. Requir. Knowledge, MARK 2009 12–21. doi:10.1109/MARK.2009.3.

[S33] Onions, P., Patel, C., 2009. Enterprise SoBA: Large-scale implementation of acceptance test driven story cards. 2009 IEEE Int. Conf. Inf. Reuse Integr. IRI 2009 105–109. doi:10.1109/IRI.2009.5211600.

[S34] Bjarnason, E., Unterkalmsteiner, M., Borg, M., Engström, E., 2015. Multi-case study of agile requirements engineering and the use of test cases as requirements. Inf. Softw. Technol. doi:10.1016/j.infsof.2016.03.008.

[S35] Silva, T. R., 2016. Definition of a behavior-driven model for requirements specification and testing of interactive systems. In Requirements Engineering Conference (RE), 2016 IEEE 24th International (pp. 444–449). IEEE. doi:10.1109/RE.2016.12.

[S36] Hotomski, S., Charrada, E. Ben, Glinz, M., 2016. An Exploratory Study on Handling Requirements and Acceptance Test Documentation in Industry 116–125. doi:10.1109/RE.2016.37.

[S37] Rahayu, P., Sensuse, D.I., Fitriani, W.R., Nurrohmah, I., Mauliadi, R., Rochman, H.N., 2016. Applying usability testing to improving Scrum methodology in develop assistant information system. 2016 Int. Conf. Inf. Technol. Syst. Innov. 1–6. doi:10.1109/ICITSI.2016.785822.

[S38] Sfetsos, P., Angelis, L., Stamelos, I., Raptis, P., 2016. Integrating user-centered design practices into agile Web development: A case study. 2016 7th Int. Conf. Information, Intell. Syst. Appl. 1–6. doi:10.1109/IISA.2016.778542.

[S39] Camacho, C.R., Marczak, S., Cruzes, D., 2016. Agile Team Members Perceptions on Non-Functional Testing Influencing Factors from an Empirical Study 582–589. doi:10.1109/ARES.2016.98.

[S40] Inayat, I., Salim, S.S., 2015. A framework to study requirements-driven collaboration among agile teams: Findings from two case studies. Comput. Human Behav. 51, Part B, 1367–1379. doi:10.1016/j.chb.2014.10.040.

[S41] Inayat, I., Salim, S.S., Kasirun, Z.M., 2012. Socio-technical aspects of requirements-driven collaboration (RDC) in agile software development methods. 2012 IEEE Conf. Open Syst. ICOS 2012. doi:10.1109/ICOS.2012.6417644.

[S42] Inayat, I., Marczak, S., Salim, S.S., 2013. Studying relevant socio-technical aspects of requirements-driven collaboration in agile teams. 2013 3rd Int. Work. Empir. Requir. Eng. Emp. 2013 - Proc. 32–35. doi:10.1109/EmpiRE.2013.6615213.

[S43] Korkala, M., Abrahamsson, P., 2007. Communication in Distributed Agile Development: A Case Study. Softw. Eng. Adv. Appl. 2007. 33rd EUROMICRO Conf. 203–210. doi:10.1109/EUROMICRO.2007.23.

[S44] Abdullah, N.N.B., Honiden, S., Sharp, H., Nuseibeh, B., Notkin, D., 2011. Communication patterns of agile requirements engineering. Proc. 1st Work. Agil. Requir. Eng. - AREW '11 1–4. doi:10.1145/2068783.2068784.

[S45] Eliasson, U., Heldal, R., Knauss, E., Pelliccione, P., 2015. The need of complementing plan-driven requirements engineering with emerging communication: Experiences from Volvo Car Group. 2015 IEEE 23rd Int. Requir. Eng. Conf. RE 2015 - Proc. 372–381. doi:10.1109/RE.2015.7320454.

[S46] Bjarnason, E.., Sharp, H.., 2015. The role of distances in requirements communication: a case study. Requir. Eng. 22, 1–26. doi:10.1007/s00766-015-0233-3.

[S47] Kovitz, B., 2003. Hidden skills that support phased and agile requirements engineering. Requir. Eng. 8, 135–141. doi:10.1007/s00766-002-0162-9.

[S48] Bjarnason, E., Wnuk, K., Regnell, B., 2012. Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering. Inf. Softw. Technol. 54, 1107–1124. doi:10.1016/j.infsof.2012.04.006.

[S49] Heikkilä, V.T., Paasivaara, M., Lasssenius, C., Damian, D., Engblom, C., 2017. Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson. Empir. Softw. Eng. doi:10.1007/s10664-016-9491-z.

[S50] Dragicevic, S., Celar, S., Novak, L., 2014. Use of method for elicitation, documentation, and validation of software user requirements (MEDoV) in agile software development projects. Proc. - 6th Int. Conf. Comput. Intell. Commun. Syst. Networks, CICSyN 2014 65–70. doi:10.1109/CICSyN.2014.27.

[S51] Lucassen, G., Dalpiaz, F., Brinkkemper, S., van der Werf, J.M.E.M., 2015. Forging High-Quality User Stories: Towards a Discipline for Agile Requirements. Proc. IEEE Int. Requir. Eng. Conf. 126–135. doi:10.1109/RE.2015.7320415.

[S52] Dumas-Monette, J.F., Trudel, S., 2014. Requirements engineering quality revealed through functional size measurement: An empirical study in an agile context. Proc. - 2014 Jt. Conf. Int. Work. Softw. Meas. IWSM 2014 Int. Conf. Softw. Process Prod. Meas. Mensura 2014 222–232. doi:10.1109/IWSM.Mensura.2014.43.

[S53] Heck, P., Zaidman, A., 2015. Quality criteria for just-in-time requirements: Just enough, just-in-time? 1st Int. Work. Just-in-Time Requir. Eng. JIT RE 2015 - Proc. 1–4. doi:10.1109/JITRE.2015.733017.

[S54] Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M., Brinkkemper, S., 2016. Improving agile requirements: the Quality User Story framework and tool. Requir. Eng. 21, 383–403. doi:10.1007/s00766-016-0250-x.

[S55] Heck, P., Zaidman, A., 2016. A framework for quality assessment of just-in-time requirements: the case of open source feature requests. Requir. Eng. 1–21. doi:10.1007/s00766-016-0247-5.

[S56] Marfa, P., Kalinowski, M., Mendez Fernandez, D., Wagner, S., 2016. Towards Guidelines for Preventing Critical Requirements Engineering Problems. Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA),. doi:10.1109/SEAA.2016.50.

[S57] Heck, P., Zaidman, A., 2014. A Quality Framework for Agile Requirements: A Practitioner's Perspective. CoRR, vol. abs/1406.4692, 2014, [onlineX] Available: http://arxiv.org/abs/1406.4692.

[S58] Gebhart, M., Giessler, P., Burkhardt, P., Abeck, S., 2015. Quality-Oriented Requirements Engineering of RESTful Web Service for Systemic Consenting. International Journal on Advances in Software, vol. 8, 156–166, Available: http://www.iariajournals.org/software/.

[S59] Rubin, E., Rubin, H., 2011. Supporting agile software development through active documentation. Requir. Eng. 16, 117–132. doi:10.1007/s00766-010-0113-9.

[S60] Fægri, T.E., Moe, N.B., 2015. Re-conceptualizing Requirements Engineering: Findings from a Large-scale, Agile Project. Sci. Work. Proc. XP2015 4:1–4:5. doi:10.1145/2764979.2764983.
[S61] Nawrocki, J., Jasiński, M., Walter, B., Wojciechowski, A., 2002. Extreme programming modified: Embrace requirements engineering practices. Proc. IEEE Int. Conf. Requir. Eng. 2002-Janua, 303–310. doi:10.1109/ICRE.2002.1048543.
[S62] Gallardo-Valencia, R.E., Olivera, V., Sim, S.E., 2007. Are Use Cases beneficial for developers using Agile Requirements? 2007 5th Int. Work. Comp. Eval. Requir. Eng. CERE 11–21. doi:10.1109/CERE.2007.2.
[S63] Stettina, C.J., Kroon, E., 2013. Is There an Agile Handover? An Empirical Study of Documentation and Project Handover Practices Across Agile Software Teams. 19th ICE IEEEITMC Int. Conf. Hague, Netherlands.
[S64] López-Nores, M., Pazos-Arias, J.J., García-Duque, J., Barragáns-Martínez, B., 2006. An agile approach to support incremental development of requirements specifications. Proc. Aust. Softw. Eng. Conf. ASWEC 2006, 9–18. doi:10.1109/ASWEC.2006.11.
[S65] Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M., Brinkkemper, S., 2016. The use and effectiveness of user stories in practice. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9619, 205–222. doi:10.1007/978-3-319-30282-9_14.
[S66] Araújo, J., Ribeiro, J.C., 2005. Towards an aspect-oriented agile requirements approach. Int. Work. Princ. Softw. Evol. 2005, 140–143. doi:10.1109/IWPSE.2005.31.
[S67] Kumar, M., Shukla, M., Agarwal, S., 2013. A Hybrid Approach of Requirement engineering in Agile Software Development. 2013 Int. Conf. Mach. Intell. Res. Adv. 515–519. doi:10.1109/ICMIRA.2013.108.
[S68] Käpyaho, M., Kauppinen, M., 2015. Agile Requirements Engineering with Prototyping: A Case Study. IEEE Int. Requir. Eng. Conf. 23, 334–343. doi:10.1109/RE.2015.7320450.
[S69] Mahmud, I., Veneziano, V., 2011. Mind-mapping: An effective technique to facilitate requirements engineering in agile software development. 14th Int. Conf. Comput. Inf. Technol. ICCIT 2011 157–162. doi:10.1109/ICCITechn.2011.6164775.
[S70] Nytrø, Ø., Søfrby, I.D., Karpati, P., 2009. Query-based requirements engineering for health care information systems: Examples and prospects. Proc. 2009 ICSE Work. Softw. Eng. Heal. Care, SEHC 2009 62–72. doi:10.1109/SEHC.2009.5069607.
[S71] Dai, L.P., Huang, L.J., 2010. Requirements acquirement based on two-world simulation method. ICCSE 2010 - 5th Int. Conf. Comput. Sci. Educ. Final Progr. B. Abstr. 1865–1867. doi:10.1109/ICCSE.2010.5593814.
[S72] Lombriser, P., Dalpiaz, F., Lucassen, G., Brinkkemper, S., 2016. Gamified requirements engineering: Model and experimentation. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9619, 171–187. doi:10.1007/978-3-319-30282-9_12.
[S73] Alaa, G., Samir, Z., 2015. A multi-faceted roadmap of requirements traceability types adoption in SCRUM: An empirical study. 2014 9th Int. Conf. Informatics Syst. INFOS 2014 SW1–SW9. doi:10.1109/INFOS.2014.7036675.
[S74] Martakis, A., Daneva, M., 2013. Handling requirements dependencies in agile projects: A focus group with agile software development practitioners. Proc. - Int. Conf. Res. Challenges Inf. Sci. doi:10.1109/RCIS.2013.6577679.
[S75] Woit, D.M.: Requirements interaction management in an extreme programming environment: a case study. In: ICSE 2005: Proceedings of the 27th international conference on Software engineering, pp. 489–494 (2005). doi:10.1109/ICSE.2005.1553594.
[S76] Heck, P., Zaidman, A., 2014. Horizontal Traceability for Just-in-time Requirements: The Case for Open Source Feature Requests. J. Softw. Evol. Process 26, 1280–1296. doi:10.1002/smr.1678.
[S77] Alsalemi, A.M., Yeoh, E.T., 2016. A survey on product backlog change management and requirement traceability in agile (Scrum). 2015 9th Malaysian Softw. Eng. Conf. MySEC 2015 189–194. doi:10.1109/MySEC.2015.7475219.
[S78] Gayer, S., Herrmann, A., Keuler, T., Riebisch, M., Antonino, P.O., 2016. Lightweight Traceability for the Agile Architect. Computer (Long. Beach. Calif). 49, 64–71. doi:10.1109/MC.2016.150.
[S79] Furtado, F., Zisman, A., 2016. Trace++: A Traceability Approach to Support Transitioning to Agile Software Engineering. doi:10.1109/RE.2016.47.
[S80] Wanderley, F., Silva, A., Araujo, J., Silveira, D.S., 2014. SnapMind: A framework to support consistency and validation of model-based requirements in agile development. 2014 IEEE 4th Int. Model. Requir. Eng. Work. MoDRE 2014 - Proc. 47–56. doi:10.1109/MoDRE.2014.6890825.
[S81] Ribeiro, J.C., Araujo, J., 2008. AspOrAS: A requirements agile approach based on scenarios and aspects. Proc. 2nd Int. Conf. Res. Challenges Inf. Sci. RCIS 2008 313–323. doi:10.1109/RCIS.2008.4632121.
[S82] T. Tenso, A. Norta, and I. Vorontsova. Evaluating a novel agile requirements engineering method: A case study. In Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering - Volume 1: ENASE, pages 156–163, 2016.
[S83] Rivero, J.M., Grigera, J., Distante, D., Montero, F., Rossi, G., 2017. DataMock: An Agile Approach for Building Data Models from User Interface Mockups. Softw. Syst. Model. doi:10.1007/s10270-017-0586-9.
[S84] Wautelet, Y., Heng, S., Kolp, M., Mirbel, I., Poelmans, S., 2016. Building a rationale diagram for evaluating user story sets. Proc. - Int. Conf. Res. Challenges Inf. Sci. 2016–August. doi:10.1109/RCIS.2016.7549299.
[S85] Furfaro, A., Gallo, T., Garro, A., Saccà, D., & Tundis, A. 2016. ResDevOps: A Software Engineering Framework for Achieving Long-Lasting Complex Systems. In Requirements Engineering Conference (RE), 2016 IEEE 24th International (pp. 246–255). IEEE. doi:10.1109/RE.2016.15.
[S86] Cao, L., Ramesh, B., 2008. Agile requirements engineering practices: An empirical study. IEEE Softw. 25, 60–67. doi:10.1109/MS.2008.1.
[S87] Bjarnason, E., Wnuk, K., Regnell, B., 2011. A Case Study on Benefits and Side-effects of Agile Practices in Large-scale Requirements Engineering, in: Proceedings of the 1st Workshop on Agile Requirements Engineering, AREW '11. ACM, New York, NY, USA, p. 3:1–3:5. doi:10.1145/2068783.2068786.
[S88] Rolland, K.H., 2015. "Desperately" Seeking Research on Agile Requirements in the Context of Large-Scale Agile Projects. Proc. XP2015. doi:10.1145/2764979.2764984.
[S89] Kassab, M., 2014. An Empirical Study on the Requirements Engineering Practices for Agile Software Development. 2014 40th EUROMICRO Conf. Softw. Eng. Adv. Appl. 254–261. doi:10.1109/SEAA.2014.77.

[S90] Ernst, N.A., Borgida, A., Jureta, I.J., Mylopoulos, J., 2014. Agile requirements engineering via paraconsistent reasoning. Inf. Syst. 43, 100–116. doi:10.1016/j.is.2013.05.008.
[S91] Macasaet, R., Chung, L., Garrido, J.L., Noguera, M., Rodríguez, M.L., 2011. An Agile Requirements Elicitation Approach Based on {NFRs} and Business Process Models for Micro-businesses. Proc. 12th Int. Conf. Prod. Focus. Softw. Dev. Process Improv. 50–56. doi:10.1145/2181101.2181114.
[S92] Sen, A.M., Hemachandran, K., 2010. Elicitation of goals in requirements engineering using agile methods. Proc. - Int. Comput. Softw. Appl. Conf. 263–268. doi:10.1109/COMPSACW.2010.53.
[S93] Kumar, M., Ajmeri, N., Ghaisas, S., 2010. (ACM-0057) Towards knowledge assisted agile requirements evolution. RSSE '10 Proc. 2nd Int. Work. Recomm. Syst. Softw. Eng. 16–20. doi:10.1145/1808920.1808924.
[S94] Butt, S.M., Ahmad, W.F.W., 2012. Handling requirements using FlexREQ model. ICSESS 2012 - Proc. 2012 IEEE 3rd Int. Conf. Softw. Eng. Serv. Sci. 661–664. doi:10.1109/ICSESS.2012.6269553.
[S95] Anitha, P.C., Savio, D., Mani, V.S., 2013. Managing requirements volatility while "Scrumming" within the V-Model. 2013 3rd Int. Work. Empir. Requir. Eng. Emp. 2013 - Proc. 17–23. doi:10.1109/EmpiRE.2013.6615211.
[S96] Thamrongchote, C., Vatanawood, W., 2016. Business process ontology for defining user story. 2016 IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. ICIS 2016 - Proc. 3–6. doi:10.1109/ICIS.2016.7550829.
[S97] Grapenthin, S., Book, M., Richter, T., Gruhn, V., 2016. Supporting Feature Estimation with Risk and Effort Annotations. 2016 42th Euromicro Conf. Softw. Eng. Adv. Appl. 17–24. doi:10.1109/SEAA.2016.24.
[S98] Sillitti, A., Ceschi, M., Russo, B., Succi, G., 2005. Managing uncertainty in requirements: A survey in documentation-driven and Agile companies. Proc. - Int. Softw. Metrics Symp. 2005, 145–154. doi:10.1109/METRICS.2005.29.
[S99] Kropp, E., Koischwitz, K., 2014. User-centered-design in agile RE through an On-site User Experience Consultant. 2014 IEEE 2nd Int. Work. Usability Access. Focus. Requir. Eng. UsARE 2014 - Proc. 9–12. doi:10.1109/UsARE.2014.6890994.
[S100] Dragicevic, S., Celar, S., Turic, M., 2017. Bayesian network model for task effort estimation in agile software development. J. Syst. Softw. 127, 109–119. doi:10.1016/j.jss.2017.01.027.
[S101] Ochodeck M., 2016. Approximation of COSMIC functional size of scenario-based requirements in Agile based on syntactic linguistic features — a replication study. 2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement. doi:10.1109/IWSM-Mensura.2016.8.
[S102] Dimitrijević S., Jovanovic, J., Devedžić,V., 2015. A comparative study of software tools for user story management. Inf. Softw. Technol. 57, 352–368. doi:10.1016/j.infsof.2014.05.012.
[S103] Soundararajan, S., Arthur, J.D., 2009. A soft-structured agile framework for larger scale systems development. Proc. Int. Symp. Work. Eng. Comput. Based Syst. 187–195. doi:10.1109/ECBS.2009.21.
[S104] Voigt, S., Huettemann, D., Gohr, A., 2016. sprintDoc: Concept for an Agile Documentation Tool. 2016 11Th Iber. Conf. Inf. Syst. Technol. 1–4. doi:10.1109/CISTI.2016.7521550.

Appendix B. Distribution of selected studies

Id.	Publication	Acronym	Type
1	11th Iberian Conference on Information Systems and Technologies	CISTI	Conference
2	11th International Conference on Availability, Reliability and Security	ARES	Conference
3	11th International Conference on Evaluation of Novel Software Approaches to Software Engineering	ENASE	Conference
4	**42th Euromicro Conference on Software Engineering and Advanced Applications**	**SEAA**	**Conference**
5	**42th Euromicro Conference on Software Engineering and Advanced Applications**	**SEAA**	**Conference**
6	7th International Conference on Information, Intelligence, Systems & Applications	IISA	Conference
7	9th Malaysian Software Engineering Conference	MySEC	Conference
8	ACIS International Conference on Software Engineering Research, Management and Applications	SERA	Conference
9	Agile Conference	AGILE	Conference
10	Asia-Pacific Software Engineering Conference	APSEC	Conference
11	Australian Software Engineering Conference	ASWEC	Conference
12	**EUROMICRO Conference on Software Engineering and Advanced Applications**	**SEAA**	**Conference**
13	**EUROMICRO Conference on Software Engineering and Advanced Applications**	**SEAA**	**Conference**
14	**EUROMICRO Conference on Software Engineering and Advanced Applications**	**SEAA**	**Conference**
15	Hawaii International Conference on System Sciences	HICSS	Conference
16	**IEEE 24th International Requirements Engineering Conference**	**RE**	**Conference**
17	**IEEE 24th International Requirements Engineering Conference**	**RE**	**Conference**
18	**IEEE 24th International Requirements Engineering Conference**	**RE**	**Conference**
19	**IEEE 24th International Requirements Engineering Conference**	**RE**	**Conference**
20	IEEE 24th International Requirements Engineering Conference Workshops	REW	Conference
21	IEEE Conference on Open Systems	ICOS	Conference
22	IEEE International Conference and Workshop on the Engineering of Computer Based Systems	ECBS	Conference
23	IEEE International Conference on Information Reuse & Integration	IRI	Conference
24	IEEE International Conference on Software Engineering and Service Science	ICSESS	Conference
25	**IEEE International Requirements Engineering Conference**	**RE**	**Conference**
26	**IEEE International Requirements Engineering Conference**	**RE**	**Conference**
27	**IEEE International Requirements Engineering Conference**	**RE**	**Conference**
28	**IEEE International Requirements Engineering Conference**	**RE**	**Conference**
29	**IEEE International Requirements Engineering Conference**	**RE**	**Conference**

(*continued on next page*)

Id.	Publication	Acronym	Type
30	**IEEE Southeastcon**		**Conference**
31	**IEEE Southeastcon**		**Conference**
32	**IEEE Southeastcon**		**Conference**
33	**IEEE Southeastcon**		**Conference**
34	**IEEE Southeastcon**		**Conference**
35	IEEE/ACIS 15th International Conference on Computer and Information Science	ICIS	Conference
36	IEEE/ACM International Conference on Automated Software Engineering	ASE	Conference
37	International Conference on Advanced Communication Technology	ICACT	Conference
38	International Conference on Computational Intelligence, Communication Systems and Networks	CICSYN	Conference
39	International Conference on Computer and Information Technology	ICCIT	Conference
40	International Conference on Computer Science and Education	ICCSE	Conference
41	International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering	ICCNEEE	Conference
42	International Conference on Engineering, Technology and Innovation	ICE	Conference
43	International Conference on Informatics and Systems	INFOS	Conference
44	International Conference on Information Technology Systems and Innovation	ICITSI	Conference
45	International Conference on Issues and Challenges in Intelligent Computing Techniques	ICICT	Conference
46	International Conference on Machine Intelligence and Research Advancement	ICMIRA	Conference
47	International Conference on Online Communities and Social Computing	OCSC	Conference
48	International Conference on Product Focused Software Development and Process Improvement	Profes	Conference
49	**International Conference on Research Challenges in Information Science**	**RCIS**	**Conference**
50	**International Conference on Research Challenges in Information Science**	**RCIS**	**Conference**
51	**International Conference on Research Challenges in Information Science**	**RCIS**	Conference
52	**International Conference on Research Challenges in Information Science**	**RCIS**	**Conference**
53	International Conference on Software Engineering	**ICSE**	Conference
54	**International Working Conference on Requirements Engineering: Foundation for Software Quality**	**REFSQ**	**Conference**
55	**International Working Conference on Requirements Engineering: Foundation for Software Quality**	**REFSQ**	**Conference**
56	**International Working Conference on Requirements Engineering: Foundation for Software Quality**	**REFSQ**	**Conference**
57	Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement	IWSM-MENSURA	Conference
58	Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement	IWSM-MENSURA	Conference
59	Malaysian Software Engineering Conference	MySEC	Conference
60	The Ninth International Conference on Software Engineering Advances	ICSEA	Conference
61	arXiv		Tech. Report
62	Computers in Human Behavior		Journal
63	**Empirical Software Engineering**		**Journal**
64	**Empirical Software Engineering**		**Journal**
65	**Empirical Software Engineering**		**Journal**
66	**Information and Software Technology**		**Journal**
67	**Information and Software Technology**		**Journal**
68	**Information and Software Technology**		**Journal**
69	**Information and Software Technology**		**Journal**
70	Information Systems		Journal
71	J. Softw. Evol. Process		Journal
72	Journal of Emerging Technologies in Web Intelligence		Journal
73	Journal of Systems and Software		Journal
74	Journal of Systems and Software		Journal
75	**Requirements Engineering**		**Journal**
76	**Requirements Engineering**		**Journal**
77	**Requirements Engineering**		**Journal**
78	**Requirements Engineering**		**Journal**
79	**Requirements Engineering**		**Journal**
80	**Requirements Engineering**		**Journal**
81	**Requirements Engineering**		**Journal**
82	Software & Systems Modeling		Journal
83	Computer		Magazine
84	IEEE Software		Magazine
85	ACM-IEEE International Symposium on Empirical Software Engineering and Measurement	ESEM	Symposium
86	IEEE International Symposium Software Metrics		Symposium
87	Annual Computer Software and Applications Conference Workshops	COMPSACW	Workshop
88	IEEE International Model-Driven Requirements Engineering Workshop	MoDRE	Workshop
89	IEEE Workshop on Just-In-Time Requirements Engineering (JITRE)	JITRE	Workshop
90	International Workshop on Comparative Evaluation in Requirements Engineering	CERE	Workshop
91	International Workshop on Empirical Requirements Engineering	EmpiRE	Workshop
92	International Workshop on Empirical Requirements Engineering	EmpiRE	Workshop
93	International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises	WETICE	Workshop
94	International Workshop on Principles of Software Evolution	IWPSE	Workshop
95	International Workshop on Recommendation Systems for Software Engineering	RSSE	Workshop
96	International Workshop on Software Product Management	IWSPM	Workshop
97	International Workshop on Usability and Accessibility Focused Requirements Engineering	UsARE	Workshop
98	**Scientific Workshop Proceedings of the XP**	**XP**	**Workshop**
99	**Scientific Workshop Proceedings of the XP**	**XP**	**Workshop**
100	**Scientific Workshop Proceedings of the XP**	**XP**	**Workshop**
101	Second International Workshop on Managing Requirements Knowledge	MARK	Workshop
102	Workshop on Agile Requirements Engineering	AREW	Workshop
103	Workshop on Agile Requirements Engineering	AREW	Workshop
104	Workshop on Software Engineering in Health Care	ICSE - SEHC	Workshop

References

Dybå, T., Dingsøyr, T., 2008. Empirical studies of agile software development: a systematic review. Inf. Softw. Technol. 50, 833–859. doi:10.1016/j.infsof.2008.01.006.

Diebold, P., Dahlem, M., 2014. Agile practices in practice: a mapping study. In: Proc. 18th Int. Conf. Eval. Assess. Softw. Eng. (EASE '14), 30, p. 10. doi:10.1145/2601248.2601254.

Agile Manifesto: Manifesto for Agile Software Development. 2001 17 February. Available: <http://www.agilemanifesto.org/>. Visited in: 11/02/2016.

Standish Group –"The Chaos Report 2014. Available: <http://www.projectsmart.co.uk/white-papers/chaos-report.pdf>.

Kotonya, G., Sommerville, I., 1997. Requirements Engineering. John Wiley & Sons.

Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B., 2012. A decade of agile methodologies: Towards explaining agile software development. J. Syst. Softw. 85, 1213–1221. doi:10.1016/j.jss.2012.02.033.

Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S., 2015. A systematic literature review on agile requirements engineering practices and challenges. Comput. Hum. Behav. 51. doi:10.1016/j.chb.2014.10.046.

Kitchenham, B.A, Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Keele University and Durham University Technical Report EBSE 2007-001, Version 2.3.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering, 17, p. 1.

Sommerville, I., 2001. Software Engineering, 6th ed. Addison-Wesley, Harlow, England; New York.

K. Schwaber, J. Sutherland (2018) The *Scrum* guide - The *definitive guide to Scrum*: the rules of the game, https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100. Accessed 14 December.

Bano, M., Zowghi, D., Ikram, N., 2014. Systematic reviews in requirements engineering: a tertiary study. Empirical Requirements Engineering (EmpiRE) at RE 2014 Sweden 2014.

Medeiros, J.D.R.V, Alves, D.C.P., Vasconcelos, A., Silva, C., Wanderley, E., 2015. Requirements engineering in agile projects: a systematic mapping based in evidences of industry. In: CIBSE 2015 - XVIII Ibero-American Conf. Softw. Eng, pp. 460–473.

Heikkila, V.T., Damian, D., Lassenius, C., Paasivaara, M., 2015. A mapping study on requirements engineering in agile software development. In: 2015 41st Euromicro Conf. Softw. Eng. Adv. Appl., pp. 199–207. doi:10.1109/SEAA.2015.70.

Schön, E.-M., Thomaschewski, J., Escalona, M.J. Agile Requirements Engineering: A Systematic Literature Review. Comput. Stand. Interfaces http://dx.doi.org/10.1016/j.csi.2016.08.011.

Eghariani, K., Kama, N., 2016. Review on Agile requirements engineering challenges. In: 2016 3rd Int. Conf. Comput. Inf. Sci., pp. 507–512. doi:10.1109/ICCOINS.2016.7783267.

Heck, P., Zaidman, A., 2016. A systematic literature review on quality criteria for agile requirements specifications. Software Quality Journal. Springer, US doi:10.1007/s11219-016-9336-4.

Medeiros, J., Goulao, M., Vasconcelos, A., Silva, C., 2016. Towards a model about quality of software requirements specification in agile projects. In: 2016 10th Int. Conf. Qual. Inf. Commun. Technol, pp. 236–241. doi:10.1109/QUATIC.2016.058.

Magües, D.A., Castro, J.W., Acuña, S.T., 2016. Usability in agile development: a systematic mapping study. In: Proc. XLII Conf. Latinoam. Informaťica doi:10.1109/CLEI.2016.7833347.

Silva, T., Martin, A., Maurer, F., Silveira, M., 2011. User-centered design and agile methods: a systematic review. In: Agil. Conf. (AGILE), 2011, pp. 77–86. doi:10.1109/AGILE.2011.24.

Bourque, P., Fairley, R.E., 2014. Guide to the Software Engineering - Body of Knowledge (SWEBOK), Version 3.0. IEEE Computer Society doi:10.1234/12345678.

Beck, K., Andres, C., 2004. Extreme Programming Explained: Embrace Change, 2nd edition Addison-Wesley, Boston, MA, USA.

Schwaber, K., Beedle, M., 2002. Agile Software Development with Scrum. Prentice-Hall, Upper Saddle River, NJ, USA.

Wang, X., Conboy, K., Cawley, O., 2012. "Leagile" software development: an experience report analysis of the application of lean approaches in agile software development. J. Syst. Softw. 85 (6), 1287–1299.

Wohlin, C., 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE14, 2014, p. 38. doi:10.1145/2601248.2601268.

Jalali, S., Wohlin, C., 2012. Systematic literature studies: database searches vs. backward snowballing. In: Int. Symp. Empir. Softw. Eng. Meas, pp. 29–38. doi:10.1145/2372251.2372257.

Wohlin, C., Runeson, P., Da Mota Silveira Neto, P.A., Engström, E., Do Carmo Machado, I., De Almeida, E.S., 2013. On the reliability of mapping studies in software engineering. J. Syst. Softw. 86, 2594–2610. doi:10.1016/j.jss.2013.04.076.

Landis, J.R., Kosh, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 33 (1), 159–174.

Wake, B.. INVEST in good stories, and SMART tasks http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/ accessed: 2015-07-20.

ISO/IEC/IEEE, ISO/IEC/IEEE 29148:2011 Systems and software engineering – life cycle processes – requirements engineering, 2011.

Wohlin, C., Höst, M., Henningsson, K., 2003. Empirical research methods in software engineering. Empir. Methods Stud. Softw. Eng. 2765, 7–23. doi:10.1007/b11962.

Wieringa, R., Maiden, N., Mead, N., Rolland, C., 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requir. Eng. 11, 102–107. doi:10.1007/s00766-005-0021-6.

Cao, L., Ramesh, B., 2008. Agile requirements engineering practices: an empirical study. IEEE Softw. 25, 60–67. doi:10.1109/MS.2008.1.

Karina Curcio has a Bachelor's degree in Informatics, Master's in Electrical Engineering at Federal University of Technology - Paraná. She is a PhD student and member of the Software Engineering Research Group at Pontifical Catholic University of Paraná and her research interests are in software engineering, software quality, software process improvement and innovation.

Tiago Navarro has a Bachelorś Degree in Informatics and is a Master student at the Pontifical Catholic University of Paraná. He is a professor at SENAI, teaching topics related to software engineering, software development and technologies.

Andreia Malucelli has a Bachelor's Degree in Informatics, a Master's Degree in Electrical Engineering and a Doctorate in Electrical and Computing Engineering. She is a professor at the Pontifical Catholic University of Paraná and an experienced researcher in the following areas of computer science: software engineering, artificial intelligence, organizational learning, ontologies, multiagent systems and healthcare information systems.

Sheila Reinehr has a Bachelor's Degree in Mechanical Engineering, a Master's Degree in Informatics and a Doctorate in Engineering. She is a professor at the Pontifical Catholic University of Paraná and an experienced researcher in the following areas of computer science: software engineering, software process improvement, software quality, project management, software product lines and metrics.