Agile Requirements Engineering: A systematic literature review

Eva-Maria Schön [a], Jörg Thomaschewski [b], María José Escalona [a]

[a] University of Seville, Spain
[b] University of Applied Sciences Emden/Leer, Emden, Germany

Keywords:
Systematic literature review
Agile software development
Requirements Engineering
Human-computer interaction
User-centered design

A B S T R A C T

Nowadays, *Agile Software Development* (ASD) is used to cope with increasing complexity in system development. Hybrid development models, with the integration of *User-Centered Design* (UCD), are applied with the aim to deliver competitive products with a suitable *User Experience* (UX). Therefore, stakeholder and user involvement during *Requirements Engineering* (RE) are essential in order to establish a collaborative environment with constant feedback loops. The aim of this study is to capture the current state of the art of the literature related to Agile RE with focus on stakeholder and user involvement. In particular, we 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.

We conduct a Systematic Literature Review (SLR) with an extensive quality assessment of the included studies. We identified 27 relevant papers. After analyzing them in detail, we 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). Agile RE is a complex research field with cross-functional influences. This study will contribute to the software development body of knowledge by assessing the involvement of stakeholder and user in Agile RE, providing methodologies that make ASD more human-centric and giving an overview of requirements management in ASD.

1. Introduction

Nowadays the business world is characterized bycomplexity, since market requirements are changing quickly. Accordingly, providers are facing the challenge to reduce time to market while delivering innovative products that customer love. Agile software development (ASD) promises benefits such as on-time delivery and customer satisfaction [1], thus it aims to deliver business value in short iterations. Therefore, the development process is carried out incrementally and empirically, which is an advantage because direction of product development can be changed immediately. Humans and interactions are at the center of such methodologies [2]. Agile methodologies (e.g. Scrum [3], Kanban [4] or Extreme Programming [5]) provide a process model to develop products. These models lack in defining the right kind of product, which fulfils user needs and customer expectations. In order to fill in this gap and to develop products with a good *user experience* (UX), hybrid development approaches including *Human-Centered Design* ([6] referred to as User-Centered Design, UCD) are applied. Although there are some challenges reported while integrating ASD and UCD (see 2.1), the integration of both makes development process more human-centered [7]. Stakeholder and user involvement is a critical success factor for a system to succeed [8] and, if compared with traditional approaches, this involvement is not limited to early phases of development, as stakeholder and user are involved throughout the whole development process instead [9].

Requirements are the base of all software products and consequently Requirements Engineering (RE) plays and important role in system development. Compared to traditional RE approaches ([10], [11]), a list of prioritized requirements (Product Backlog [3]) is used instead of a requirements specification document. The main RE activities (*elicitation, documentation, validation, negotiation and management*) are not clearly separated activities in *Agile RE*. They are repeated each iteration and only required information is elaborated before the next iteration starts. For this purpose, RE in Agile environments is carried out just-in-time with a Little Design Up Front [12].

This article reports the findings of a Systematic Literature Review (SLR) in the field of Agile RE with focus on stakeholder and

user involvement. In particular, ASD, RE and UCD have one thing in common: stakeholder and user involvement is described as critical success factor for a system to succeed. To this end, this will be an important aspect in this literature review and will be addressed by the following research questions:

- 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?
  In terms of RE, these research questions lead us to the third research question:
- RQ3: What are the common ways for requirements management in ASD?

The paper is structured as follows: Section 2 gives a brief overview of Agile RE context, including a gap analysis of related work. Section 3 presents our research objectives and research questions and deals with our review method covering a description of the search strategy, selection process, quality assessment, data extraction and analysis. Section 4 summarizes the key findings of our study, therefore it offers an overview of the included studies as well as answers to our three research questions. Finally, Section 5 discusses on the meaning of findings and limitations of this study.

2. Background – Agile Requirements Engineering

In the mid-80s, Takeuchi et al. [13] already stated that a sequential phases approach to product development is not well suited due to the lack of flexibility. Since then, new process models have been developed. On one hand, there are iterative process models like *Rational Unified Process* [14,15]. On the other hand, there are Agile methodologies such as Scrum [16,3], Extreme Programming (XP) [5], Feature-Driven Development [17] and Kanban. The usage of Kanban for Information Technology (IT) was mainly influenced by Anderson in between 2004–2010 [18,4].

A number of papers regarding lightweight process models have been published. In 2001, the leaders of these different streams joined together and created the Manifesto for Agile Software Development [2]. The Agile Manifesto includes values and principles that help to optimize the software development process and have also a strong influence on nowadays team collaboration within ASD [19]. The Agile Manifesto provides the four core values listed below:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Since the Agile movement, software development has moved away from plan-driven to value-driven process models (see Fig. 1). People in plan-driven environments often negotiate about pricing models, project plans and how many product features they can develop with the available resources. They are emphasizing the generated outputs (e.g. number of created features or number of releases during a time period). In contrast, people in value-driven environments discuss visions, experiences and human values as well as how they can address them through the product. They concentrate on outcomes, which means that they are focused on the difference that outputs entail. Therefore, product development with agile methodologies is mainly driven by human values.

In the context of agile methodologies, RE is carried out iteratively during the whole development process instead of during a closed phase in the beginning. To this end, a just-in-time model is often used to refine high level requirements into low level tasks that can be implemented by developers. Therefore, business people, stakeholders, users and developers work together in a collaborative manner. The model is artifact-based and starts with capturing requirements by means of epics. An epic is a large user story [20], that can be refined by utilizing story maps [21]. A story map consists of user stories or persona stories [22], which are split into tasks. The whole workflow can be managed by means of Kanban boards for design, development and delivery [23].

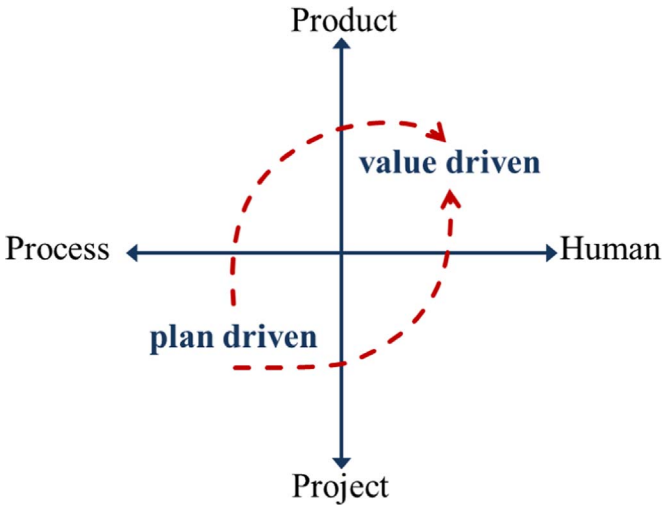

Fig. 1. In ASD people focus on outcomes and how they can fulfil human needs through the outputs they produce.

Agile techniques like Continuous Delivery (CD) [24] have an impact on the manner and the frequency of usability testing nowadays [25]. New information is given along the system development by the user and the system itself. This knowledge is processed during further steps and it conditions the decision-making process. Therefore, requirements are treated like assumptions, which are validated continuously.

There are also initiatives focusing on the alignment of RE and testing, which investigate the practice of using test cases as requirements [26]. In this context, detailed requirements are often documented as test cases, rather than using additional requirements specification, in order to reduce the effort of updating separate artifacts [26].

2.1. Summary of related literature reviews

In the literature, many reviews are conducted in order to do research on ASD. The next paragraphs briefly summarize the most related ones.

In 2011, Silva et al. [27] carried out a SLR on the integration of ASD and UCD and analyzed how usability issues are addressed in agile projects. They included a comprehensive classification process based on a system covering research-related and content-related information. The authors identified the following key aspects, which play an important role for the integration: little upfront design, prototyping, user stories, user testing, inspection evaluation and one sprint ahead. Besides, they presented a process model for the integration of ASD and UCD that took into account their findings.

Salah et al. [28] addressed a similar area. Their review aimed to identify challenging factors for the integration of ASD and UCD. They presented the challenges in a very understandable manner with good examples. Additionally, they explored practices and success factors to face these challenges. The reported challenges

are: lack of allocated time for upfront activities, difficulty of modularization, optimizing the work between developers and UCD practitioners, performing usability testing and lack of documentation.

Brhel et al.'s [29] literature review, published in 2015, examined hybrid development models, such as ASD and UCD. Their main objective was to capture the state of the art of the integration of ASD and UCD. Compared to [27] and [28], they addressed a more holistic view of ASD. Thus, they focused on four dimensions (process, people/social, technology and practices) with a coding system and five derived principles: *"(1) separate product discovery and product creation, (2) iterative and incremental design and development, (3) parallel interwoven creation tracks, (4) continuous stakeholder involvement, and (5) artifact-mediated communication"*. Furthermore, they contributed to a classification system for existing work in the field of user-centered ASD.

The main purpose of Inayat et al.'s (2015) [30] literature review is to make clear Agile RE challenges and practices, including a good discussion on related work. Moreover, they aimed to understand how traditional RE problems are resolved using Agile RE. In summary, they provided 17 commonly used practices and also practical challenges that agile teams had to face. The practices are: Face-to-face communication, customer involvement and interaction, user stories, iterative requirements, requirement prioritization, change management, cross-functional teams, prototyping, testing before coding, requirements modeling, requirements management, review meetings and acceptance tests, code refactoring, shared conceptualization, pairing for requirements analysis, retrospectives and continuous planning.

Soares et al. (2015) [31] combined a literature review with an exploratory study. They analyzed difficulties while working with requirements in an agile environment, particularly, causes that can lead to documentation debt (e.g. missing, inadequate or incomplete requirements). They contribute with their work to an important research topic, for documentation in ASD is often treated in an inadequate manner. The authors defined 10 difficulties that occur when identifying and managing agile requirements. Furthermore, they uncovered difficulties when using user stories instead of use cases.

2.2. Gap analysis

To sum up, it can be said that related literature reviews cover many aspects of Agile RE. Nevertheless, analyzing the existing work, we observed some shortcomings. Silva et al. [27] and Salah et al. [28] worked on the integration of UCD and ASD. In this context, they studied the collaboration between UCD specialists and developers but they did not pay the same attention to stakeholder and user involvement. The other published reviews [29,30,31] identified stakeholder and user involvement as one of the key aspects in ASD, although they only presented partially how this problem might be solved. As they treated this problem as one out of many, we consider that they only scratch the surface.

Since human beings and their values play one of the most important roles in value-driven organizations (see Fig. 1), it is necessary to further investigate this aspect in the Agile RE field. To this end, we conducted this SLR. However, to the best of our knowledge, no systematic review has previously been published which investigates RE with focus on stakeholder and user involvement in agile environments (see Fig. 2).

3. Review method

Appropriate guidelines have been followed for conducting the systematic review, particularly the guidelines for SLRs in Software Engineering by Kitchenham and Charters [32]. According to these guidelines, our SLR consists of three main phases. Fig. 3 shows the most important stages of each phase.

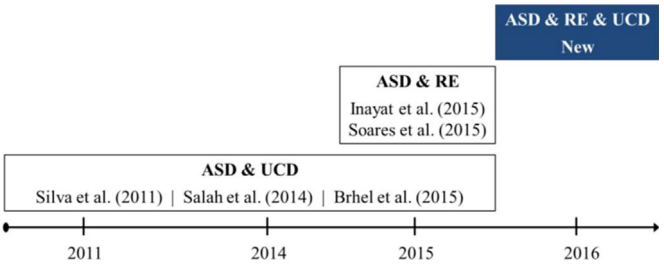

Fig. 2. Related work has not investigated Agile RE from the perspective of UCD up to now.

Due to the high number of retrieved studies, we used the software Mendeley and excel sheets in order to manage information obtained in an efficient manner.

3.1. Objectives and research questions

Our goal was to gather the state of the art of the literature related to RE, by looking at stakeholder and user involvement in agile methodologies. Therefore, we created three complementary research questions (RQ), which are specified by the following subcriteria:

3.1.1. RQ1: What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?

On one hand, we analyzed whether the existing approaches involve stakeholders and users directly into the development process. On the other hand, our aim was to study whether the approaches apply a process model for the involvement. In addition, we queried what kind of methods they use in order to gather data. With regard to the agility of the existing approaches, we examined whether there are iterations along the development process.

3.1.2. RQ2: Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?

Concerning this RQ, we analyzed the included studies in terms of methodologies that are used to handle the user perspective within agile environments. Furthermore, we investigated how the knowledge of user requirements is shared among stakeholders.

3.1.3. RQ3: What are the common ways for requirements management in ASD?

In terms of this RQ, our aim was to investigate what types of artifacts are used and how they are utilized. Moreover, we wanted to discover whether the documentation of requirements is understandable without further knowledge in order to be able to work in a collaborative manner. In addition, we examined the treatment of non-functional requirements.

3.2. Protocol development

In the beginning of the planning phase we undertook an initial informal search for other SLRs concerning a similar scope of this field. The relevant ones are presented in Section 2.1 as related works. During the informal search we found a few relevant studies, which fit our research objectives. Accompanied by the already identified studies, we used these SLRs as basis to create our RQs and to develop our review protocol [33], which was carried out in an incremental and iterative way by two independent researchers.

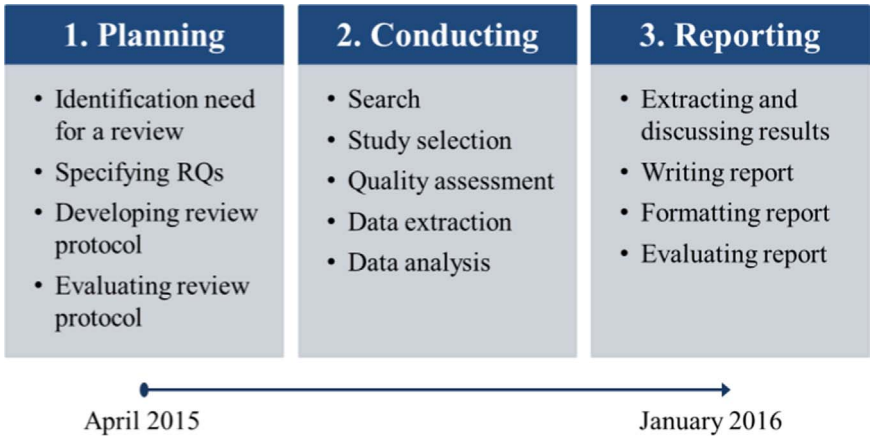

Fig. 3. Phases of a SLR.

3.3. Search strategy and data sources

Subsequent to the definition of the research objectives and the RQs we elaborated our research strategy. Therefore, we selected keywords, created a search string and specified the search space and the search process that was used to reduce the number of papers.

In a first step, we extracted a set of keywords from the studies we had and matched it with our research objectives. Secondly, we identified alternative spellings and synonyms. Since the search process is a critical aspect, we had to optimize the keywords iteratively. Thus, we defined a set of keywords, tested them in various databases and finally, we refined them. The final list can be found in Table 1.

Afterwards we connected the keywords with Boolean operators and designed our search string as follows:

(agile OR scrum OR kanban OR "extreme programming" OR lean) AND (hci OR hmi OR ucd OR usability OR human OR user) AND ("requirements engineering")

The search space included digital libraries, specific journals and conference proceedings. It is worth mentioning that every digital library has its own characteristics concerning its search engines. To this purpose, we had to adapt the search string for every library. The search was documented in a separate document that included the following information for every single digital library: name, search strategy, date of search, years covered by the search and a documentation regarding the adaption of each search string to every single trial. Table 2 shows an extract from this.

At the beginning, the search results showed a high amount of papers (42,808 findings). In order to reduce the results, we carried out the search process in different phases (see Fig. 4).

In addition to the initial search process, we started snowballing for identified papers at P6 (see Fig. 4). We applied forward snowballing (search in papers that cited the paper) and backward snowballing (search in the reference list of the paper) [34].

Snowballing helped us to identify a total amount of 965 more papers (forward snowballing N = 355; backward snowballing N = 610). For these papers, we also used the search process and started at P3. At the end of this second search process, we identified nine papers that were taken into account for data extraction.

Table 1
Keywords used for search.

Category	Keywords
Agile methodology	agile, scrum, kanban, extreme programming, lean
Human Computer Interaction	hci, hmi, ucd, usability, human, user
Requirements Engineering	requirements engineering

Table 2
Search space.

Digital library	Search strategy	Date of search
Google scholar	full text	2015-06-13
Science direct	abstract, title and keywords	2015-06-10
SpringerLink	abstract and keywords	2015-06-12
Scopus	abstract, title and keywords	2015-06-10
IEEEXplore	abstract and keywords	2015-06-12
ACM	abstract and keywords	2015-06-11

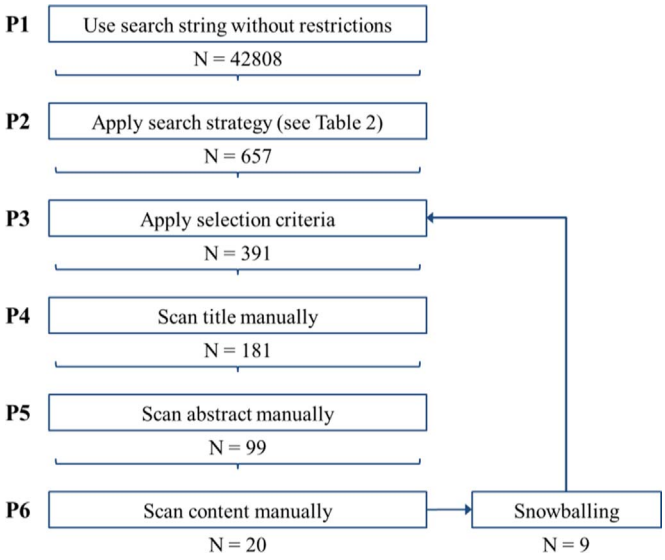

Fig. 4. Search process comprising six phases and snowballing.

3.4. Study selection

The selection criteria were divided into inclusion and exclusion criteria and were applied to P3 of the search process.

Inclusion criteria were: papers written in English; papers published in between 1995-2015; papers under peer review; papers presenting approaches to integrate user into agile development processes; papers related to Agile RE; papers associated with agile requirements documentation; and specific book chapters.

Exclusion criteria are: no full books; papers whose full text were

not available; papers with results that had been already published; papers that were not focused on agile development; papers only presenting ideas, lessons learnt, recommendations or guidelines; papers introducing tools whose underlying methodology was not comprehensibly described (black box); and studies, whose primary focus moved away from agile methodologies.

Due to the high amount of findings at P3, we reduced the time period for including papers from 1995-2015 to 2007-2015. Our aim was to cope with the current state of the art and not analyzing the evolvement over time. At the end of data extraction we found 19 papers that were obviously relevant to our study and 9 where we proofed whether they included relevant information for answering our RQs. If a paper contained relevant information, we would include it in the study. In light of this, 8 papers were only useful for the study. The other one missed relevant details, so that we discarded it during the data extraction phase.

As a result of the snowballing process, we identified a few authors, who published more than one relevant publication. Therefore, we had to identify the relevant papers we aimed to include in our study. For this purpose, we contacted the authors to either include the most cited paper or the latest one dealing with the approach.

3.5. Quality assessment

We elaborated a quality checklist to assess the individual studies. There were three available answers for every question (see Table 3). This checklist was used to evaluate the quality of the included studies.

3.6. Data extraction and analysis

According to Kitchenham and Charters' [32] guidelines, a form for data extraction was set up. We used Mendeley in order to mark text passages and ratings. That software also supported the data extraction in regard to defined attributes from the protocol:

- Basic information: title, authors, publication date, DOI and URL
- Publication data: journal, conference, date (of conference), publisher, volume, issue, pages, keywords and abstract

In addition to Mendeley, we set up a data collection form in Excel to take out the following data:

- Research method: e.g. experiment, quasi-experiment, lessons learnt,
- case study, opinion survey or tertiary study
- Research approach: deductive, inductive or hybrid
- Agile method: e.g. Scrum, XP, Kanban or hybrid
- Method: e.g. pair-programming, stand up meeting or usability pattern
- Artifacts: e.g. user stories, Kanban board, personas, prototypes
- Short summary
- Results and contributions
- Personal assessment
- Number of included references
- Number of papers that cited the study
- Ranking (CORE2014)

All identified papers were taken into consideration to carry out the data extraction process, where we found that taking out data in line with the form was not always possible because of the way studies were reported. In cases where the required information was not provided or not clearly reported we used "n.a.", in order to fill in the form. On one hand, we extracted quantitative data (e.g. publication channel or research method) and on the other hand, we extracted qualitative data (e.g. content or short summary).

We used 3-point Likert items in order to weigh the single items (*covers the criteria, covers the criteria partially or do not cover the criteria*) with the aim to answer the RQs with its sub-criteria. We chose the option *covers the criteria partially*, in cases where one item could not clearly be answered by the study.

4. Results

We included in our work 27 identified relevant studies. Firstly, we describe characteristics of the studies and show quantitative data (e.g. publication channel, research method or quality overall). Secondly, we state our findings related to the RQs.

Table 3
Quality checklist.

Item	Assessment Criteria	Score	Description
QA1	Is the proposal validated?	−1	No, it is not validated
		0	Partially, it is validated in a laboratory or only parts of the proposal are validated
		1	Yes, by a case study.
QA2	Does the study present a detailed description of the approach?	−1	No, details are missing
		0	Partially, if you want to use the approach, you need to read the references
		1	Yes, the approach can be used with presented details
QA3	Does the study present a personal opinion piece or viewpoint?	−1	Yes, it does.
		0	Partially, since related work is explained and paper is set into a specific context
		1	No, the paper is based on research
QA4	Has the study been cited by other authors?	−1	No, no one cited the study
		0	Partially, between 1-5 articles cited the study
		1	Yes, more than 5 articles cited the study
QA5	Includes the paper a clear statement of the aims of the study?	−1	No, aims are not described.
		0	Partially, aims are described but unclearly
		1	Yes, aims are well described and clear

Table 4
Distribution according to research methods.

Research Method	Paper total	Percentage
Case Study	15	56%
Case Study in laboratory	2	7%
Multi-Case Study	2	7%
Description of Approach	5	19%
Experiment	1	4%
Research Perspective	1	4%
Semi-Structured Interviews	1	4%

4.1. Summary of studies

Concerning the publication channel, the studies were published in conference proceedings or scientific journals. In comparison, 21 (78%) of the included studies were published in conference proceedings and only 6 papers (22%) appeared in scientific journals.

Table 4 presents the distribution of the studies' underlying research method. In summary, 19 studies (70%) used case studies. For this kind of publication type, we distinguished among Case Study, Multi-Case Study and Case Study in the laboratory. Most of the studies (15 paper, 56%) were carried out as single Case Study in economic enterprises. Furthermore, two works (7%) were executed in laboratories or in a simulated context. In addition, two studies (7%) were carried out as a Multi-Case Study and in five papers (19%) the authors described an approach from a theoretical viewpoint. In those papers, they did not use further research in order to validate their approaches. However, in some cases that might be a starting point for their future research. In contrast to this, results were clearly presented in one of the papers as starting point for future research activities; in consequence the authors published it as a research perspective. Furthermore, one study used an experiment and another one used semi-structured interviews as a main research method.

In conclusion, it can be stated that RE in agile methodologies is often investigated in real life context and this research field is very close to existing work practices in companies. We are aware about the fact that the results of a single case study might not be generalized to other settings and that this may have an impact on the interpretation of our results.

We used the quality checklist presented in Table 3 in order to evaluate each study. The overall results from the quality assessment are shown in Fig. 5.

The first criterion (QA1) examines whether the proposal is validated. For 15 papers this is true, as they used case studies in order to validate their proposal. Nine papers either validated their proposals in a laboratory or assessed only parts of the proposal. We also included three papers whose proposals were not validated at all. With QA2 we confirmed whether the study presented a detailed description of the approach. In 17 papers the approach is described with enough details, so that other researchers could use it. In comparison, when the approach of seven papers should be used again, included references have to be read. Two papers missed details. QA3 queried whether the study provided a personal opinion or viewpoint. 20 out of the 27 papers were based on a clearly defined research design. For 6 papers, the related work was explained and the paper was set into context. Nonetheless, there was also one study that did not clearly describe the research method. QA4 wondered how many times the study had been cited in other papers. For this purpose, we used the number of citation from Google scholar (Assessment date 2015-11-20). 15 studies had more than five citations in other papers. Six papers had been cited in among 1-5 articles and six papers had no citation until the assessment date. With the last criterion (QA5) we tested if the aims of the study were included in the paper. In 26 of the works, the aims were well described and clear. Only one paper lacked presenting the aims, since they were described very unclearly.

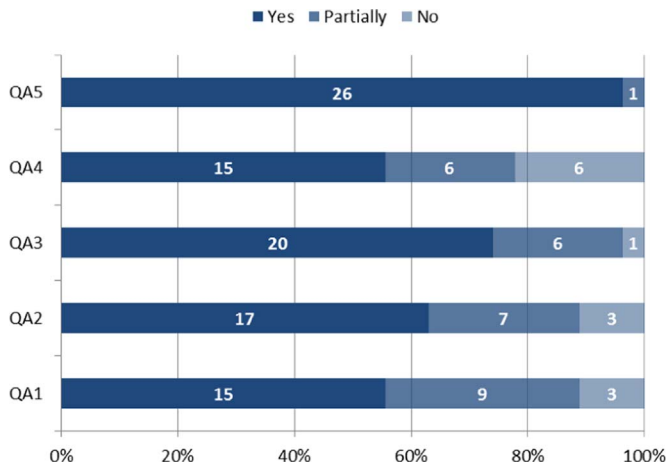

Fig. 5. Quality assessment.

To sum up, six papers fulfiled every quality criterion [35–40]. We have to be aware that the results might be different at publication date of this SLR, due to the different number of citations at assessment date that QA4 required.

4.2. (RQ 1) What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?

Table 5 presents the results from the evaluation of the subcriteria related to RQ1. Therefore, we list the studies that clearly have a positive answer, “Yes”. Additionally, the distribution according the items is also shown.

In the following paragraphs, we will put forward the highlights we found in the included studies concerning sub-criteria.

4.2.1. Stakeholder and user involvement

Bellucci et al. [41] combined XP with co-design sessions in order to develop a product with strong user involvement. They explored in a field study, how users interact and work with a prototype. Based on their findings, the prototype was developed iteratively. The authors considered this approach as a tactic to quickly deploy an evolving prototype.

Harbers et al. [42] studied the application of a *Value Story workshop* for the elicitation process of user stories. The aim of this workshop is to embed stakeholder values into the RE process. Therefore, direct and indirect stakeholders have to be identified. In a second step, the values of each stakeholder group are revealed. Then, a situation for each value is provided and stakeholder needs in this situation are analyzed.

Olsson et al. [43] derived a conceptual model from multiple case studies that emphasized the need for combining qualitative feedback in early stages of development with quantitative customer observation in later stages of development. In the *Qualitative/ Quantitative Customer-driven Development (QCD) model* requirements are treated as hypotheses that are validated through continuous customer feedback.

Several studies provided additional roles, which should be applied to an agile environment in order to address communication gap or take over responsibility for particular tasks. Dragicevic et al. [44] claimed that *business users* could collect necessary data and documents (e.g. templates, scans of documents or screens) and provide information to developers who model those data in UML.

Moreover, Lee et al. [45] stressed the role of an *Agile-UCD*

Table 5
Sub-criteria RQ1.

ID	Sub criteria	Covered by study	Overall
C1.1	Stakeholders are involved directly.	[41], [42], [43], [44], [45], [46], [47], [35], [36], [48], [49], [37], [50], [51], [52], [38], [53], [54], [39], [40], [55]	N = 27 Yes = 77% Part = 19% No = 4%
C1.2	The user is involved directly.	[41], [56], [43], [46], [47], [35], [48], [36], [49], [50], [51], [38], [39], [40], [55]	N = 27 Yes = 56% Part = 33% No = 11%
C1.3	They use a process in order to involve the stakeholders.	[41], [42], [43], [57], [44], [45], [58], [46], [47], [35], [48], [36], [49], [59], [50], [51], [52], [38], [39], [40], [55]	N = 27 Yes = 78% Part = 19% No = 4%
C1.4	There are iterations during the development process.	[41], [56], [43], [57], [44], [45], [58], [46], [47], [35], [36], [48], [49], [50], [52], [38], [39], [40], [55]	N = 27 Yes = 70% Part = 30% No = 0%
C1.5	They use methods in order to gather data.	[41], [56], [42], [43], [57], [44], [58], [60], [35], [49], [50], [59], [51], [38], [53], [54], [39], [40], [55]	N = 27 Yes = 70% Part = 30% No = 0%

specialist (AUS), which has to be responsible for bridging the communication gap between developer and UX designer. They also present a *usability-pattern-based requirement-analysis method* that help AUSs to do their tasks of requirements specification.

Kautz [38] wondered how customers and users participate in ASD with Participatory Design. He reports that customer and users were involved indirectly and directly through different kind of activities. Additionally, the role of an *onsite customer* was applied and they had weekly feedback loops. Kautz states that one of the important benefits of the frequent feedback loops is that misunderstandings are detected in time and changes can be applied early before they grow into larger problems. For this reason, the user generates a feeling of trust that has impact on the development.

Collaboration and shared understanding are essential to ASD [2]. To this end, and attending to the results of a Multi-Case Study, Ramesh et al. [39] described informal and frequent communication as the core of Agile RE. They stated that customers were directly involved in each iteration. Requirements were elicited, refined and validated through face-to-face communication with the customer. They claim requirements analysis as a social–political process that depends on human interaction and is influenced by several contextual factors.

4.2.2. Data gathering in Agile RE

The results of sub-criteria C1.5 (Table 5) revealed that 70% of the studies reviewed used methods in order to gather data. In addition to traditional methods known from agile methodologies (e.g. planning or reviews) reported by [46] and [39], the reviewed studies included further methods for data gathering within Agile RE.

For instance, we found representative examples in the following studies: Bellucci et al. [41] applied weekly co-design sessions as a meeting point between users and designers. During these sessions, researchers collected impression and feedback concerning users' experience with the system and usage scenarios. They also gathered information about non-usage and misusing of implemented features. Similarly, Kautz [38] studied *Participatory Design* activities within ASD. He reported that data gathering with customer and user involvement took place on an ongoing basis. Communication was structured through planning games, presentation of working software and acceptance tests.

Lucia et al. [47] present an overview regarding Agile RE. *Interviews, Brainstorming, Ethnography* and *Use Case analysis* are the most important elicitation techniques from their point of view.

Näkki et al. [48] describe users as a source of ideas and looked at them as decision makers throughout the design process. For this purpose, they enabled *Lead users* to participate via online co-creation tools with social media mechanisms.

Considering that analyzing the context of use is an important activity to achieve a human-centered process, several studies recommend using a structured process for this activity. They describe how to carry out a *Contextual Inquiry* [61] as an appropriate manner [40,49]. With regard to Human-Centered Design, Maguire [49] propose different methods following activities from [ISO 9241-210]. In addition to a Contextual Inquiry, he suggests to accomplish a *stakeholder analysis* to explore the context of use. Furthermore, he proposes *surveys, interviews, discussions, focus groups, competitor analysis* and *user journeys* for specifying user requirements. Maguire recommends gathering data with *conceptual design meetings* and *co-design workshops* during the creation of design solutions. For the evaluation, he highlights the use of *user walkthroughs* and *usability tests*.

4.3. (RQ 2) Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?

Table 6 shows the results from the evaluation of the sub-criteria related to RQ2. Therefore, we list the studies that clearly answered the questions positively. Moreover, we display the distribution in terms of the items.

In the following sub-sections, we will present the highlights we found in the included studies concerning sub-criteria.

4.3.1. User perspective in ASD

Cajander et al. [35] interviewed 21 IT professionals in order to analyze the user perspective in ASD. On one hand, they found that the responsibility for the user perspective is often unclear and on the other hand, they discover that in some cases the user perspective is neither discussed nor described. Furthermore, Cajander et al. state that ad hoc nature of user involvement and design feedback exist in agile projects. Moreover, they conclude that new usability methods arose because of the agile requirements (e.g. speed, efficiency or focus on deliverables instead of documentation).

4.3.2. Methodologies in Agile RE

Several studies deal with methodologies that are used to extend agile methodologies like scrum and XP, with the aim to better

Table 6
Sub criteria RQ2

ID	Sub criteria	Covered by study	Overall
C2.1	The proposal used a methodology.	[41], [42], [43], [57], [44], [58], [46], [47], [60], [48], [36], [49], [62], [59], [50], [52], [38], [53], [40], [55]	N = 25 Yes = 80 % Part = 16% No = 4%
C2.2	The knowledge about the user requirements is shared between the stakeholders.	[41], [43], [57], [45], [46], [35], [49], [48], [37], [52], [38], [53], [54], [55]	N = 27 Yes = 52% Part = 48% No = 0%

understand the user perspective.

Human-Centered Design (HCD) maguire [49] extended the HCD framework (ISO 9241-210) for agile development. There are four main activities in HCD, which are performed iteratively: a) understand and specify the context of use, b) specify user requirements, c) produce design solutions and d) evaluate designs against requirements. Maguire suggests different methods in order to perform each activity. Additionally, he recommends producing clearly defined artifacts based on the gathered information.

Design Thinking a study by Adikari et al. [60] propose a framework based on three methodologies: *Design Thinking, UX design* and *ASD*. A real world system context was being explored with other relevant systems using Design Thinking. As a result, a *reframed context* was build. They report that the knowledge of the reframed contexts could be used to create products, systems or services using UX design and ASD.

Contextual Inquiry several studies (e.g. [40] and [49]) state that performing a *Contextual Inquiry* [61] is useful to explore the user perspective and to gather data concerning both, users and context of use.

Participatory Design kautz [38] studied how customers and users participated in ASD with *Participatory Design*. Kautz focused on the role of the customers and users and how they were involved through different activities in design and development. Similarly, Bellucci et al. [41] investigated an approach to design with and by the user. For this reason, they carried out a field study to evaluate how users interacted with a prototype. Based on their findings, the prototype was developed iteratively through constant feedback loops. In addition, Näkki et al. [50] worked on an application of a lead-user approach in the context of ASD. The chosen lead users participated actively in the innovation process through idea generation as well as in all phases of the development process via online co-creation tools. With regard to Participatory Design, Olsson et al. [43] developed the *QCD* model based on a multi-case study. They confirm that it is important to combine qualitative and quantitative feedback techniques in order to achieve continuous customer validation. The authors treat requirements as hypotheses that are validated with customers before they are taken into account for development.

4.3.3. Shared understanding

Abdullah et al. [37] cope with communication patterns in an agile team and in particular, how communication and collaboration supported RE activities (gathering, clarifying and evolving) in an agile environments. They built the concept of *shared conceptualization*, which mean that the development team shares a common understanding of the requirements, which is deeper than *shared understanding*. They describe that there is a link between communication and memory. Little information about a requirement was documented on a story card, but members of the agile team built a related concept in their minds that was based on discussion concerning the requirement.

A study by Buchan [57] also provides insights into the concept of shared understanding. Buchan developed a theory of *shared understanding of requirements* (SUR). He states that SUR is a specialized form of a *Team Mental Model* with focus on RE. Furthermore, he defines two activities in SUR development: 1) uncovering a gap collaboratively, and 2) addressing this gap to achieve a new state of SUR. The goal is to enable team members achieve a consistent understanding of the requirements.

4.4. (RQ 3) What are the common ways for requirements management in ASD?

Table 7 presents the results from the evaluation of sub-criteria related to RQ3 as well as it lists the studies that clearly answered the questions positively. In addition, it also shows the distribution according the items.

In the following sub-sections, we will present the highlights we found in the included studies concerning sub-criteria.

4.4.1. Artifacts in Agile RE

Artifacts are used for communication, elaboration, validation and documentation of requirements in agile environment. In sum,

Table 7
Sub criteria RQ3.

ID	Sub criteria	Covered by study	Overall
C3.1	They are using artifacts.	[41], [56], [42], [43], [57], [44], [45], [58] [47], [46], [35], [36], [49], [37], [59], [51], [50], [52], [53], [54], [39], [38], [40], [55]	N = 27 Yes = 93% Part = 7% No = 0%
C3.2	The documentation is understandable without further knowledge	[41], [56], [42], [43], [57], [45], [46], [35], [49], [37], [59], [50], [54], [39], [38], [40], [55]	N = 27 Yes = 63% Part = 33% No = 4%
C3.3	They distinguish between functional and non-functional requirements.	[42], [44], [58], [46], [35], [36], [49], [62], [59], [52], [54], [39], [55]	N = 26 Yes = 50% Part = 42% No = 8%

Table 8
Artifacts in Agile RE.

Artifact	Description	Reference	Percentage (N=27)
User Story	User story is a description of a feature written from the perspective of the person who needs this. It consists of a written text, conversation about it and acceptance criteria.	[56], [42], [57], [45], [58], [46], [47], [48], [49], [62], [50], [38], [53], [39], [40]	56%
Prototype	Prototype is a model of the software application that supports the evaluation of design alternatives and communication.	[41], [56], [58], [46], [49], [48], [36], [50], [52], [54], [39], [40], [55]	41%
Use Case	Use case describes an action or event steps, which are needed to achieve a goal.	[42], [58], [36], [62], [59], [54], [55]	26%
Scenario	Scenario is a textual representation of a problem and describes the interaction between user and system in a specific context.	[56], [42], [58], [49], [40], [55]	22%
Story Card	Story Card is a physical representation for the written text and details from a user story.	[57], [62], [37], [38], [39]	22%
Persona	Persona is a description of a fictitious person that represents a larger part of the target group.	[56], [45], [48], [49]	15%
Vision	Vision is an abstract description of the overarching goal that guides product development and aligns development, business people and other stakeholder.	[57], [45], [55]	11%
UML diagram	Unified Modeling Language (UML) provides a standard to visualize the design of a system.	[44], [52], [53]	11%
Storyboard	With a storyboard the workflow of the user is presented in a sequence of pictures.	[50], [54], [40]	11%
Task	One user story is split into more tasks, which describe more technical requirements.	[45], [40]	7%
Kanban board	Kanban board visualizes the progress of a requirement through the workflow of the devel opment team.	[57], [45]	7%
UI pattern	UI pattern describes an abstract solution for recurring design problems and give inspiration to designer.	[53], [55]	7%
Essential use cases	Essential use case describes user tasks and is a simplified and generalized form of use cases.	[53], [55]	7%
Pictures	Picture is a visual representation (e.g. photograph, painting)	[41], [50]	7%
Videos	Video consists of a sequence of images processed electronically that are seen in a recording and displayed on a screen.	[41], [50]	7%
Mind Map	Mind map is a diagram used to visualize and organize information.	[47], [51]	7%
UI specification	Written specification that describes the UI of a system. The text is enriched by mock-ups, icons, etc.	[56], [49]	7%

we identify 57 different artifacts mentioned in the included studies. 17 out of 57 are mentioned more than twice (see Table 8).

At this point, we would like to highlight some key artifacts (usage $< 20\%$).

User stories are the most frequent used artifact in ASD. The included studies describe how they can be created and represented. Näkki et al. [50] use the concept of *needs-based user stories*. Therefore, they collected users' everyday needs and challenges regarding a specific domain. Users were involved during requirements elaboration, by commenting and rating features in order to allow the prioritization of features. Harbers et al. [42] suggest using a *Value Story workshop* to embed stakeholder values in the elicitation process of requirements. The requirements resulting from this workshop are collected as *value-based user stories*. In addition to the classical format of user stories [20], Wanderley et al. [47] present a visual language for user stories. The visual representation of the user story supports the evaluation of the requirements with users and can be utilized with a *User Story Visual Editor*.

Prototypes are categorized by the studies on different types of *fidelity* (low, mid and high fidelity). Besides, the authors use the terms *prototypes, mock-ups* and *wireframes*. Lucia et al. [54] recommend using *paper prototypes* to document requirements with the purpose of communication and knowledge sharing between stakeholders and agile teams. In addition, Obendorf et al. [40] state that paper prototypes and informal drawings are very useful in discussions with users. Informal drawings (sketches) were also used in Blomkvist et al. [56]. In comparison, Rivero et al. [46] applied html mockups to start the modeling process in Model-Driven Web Engineering. HTML-based mockups were used, on one hand, as a foundation to specify features like content, navigation and business logic and, on the other hand, to generate platform-independent UI specifications. Furthermore, Nawrocki et al. [58] propose to use mockups to elicit test cases from users to make use cases testable.

Use Cases (UC) are often used to describe the behavior of a system from a more technical viewpoint compared to user stories. Issa et al. [59] built a *UC patterns catalogue* that could be used as a feature checklist and to design an initial version of a UC model. Their *UC meta-model* addresses the environmental, technical, structural, eventual and traceability dimensions of the anticipated system. Besides, Nawrocki et al. [58] argued that UCs could be reused for user manual generation, and for effort estimation. Farid [62] divides requirements into functional and non-functional. Functional requirements are described as *Agile Use Case,* whereas non-functional requirements are presented as *Agile Loose Case. Aspect-Oriented "pointcut" operators* link functional to non-functional requirements.

Scenarios describe how users interact with a system in a specific context. To this end, they are often combined with *personas* [49,56]. In Bellucci et al. [41], designers used a journal in order to document scenarios based on user insights, gathered with a *user diary* and *usage try-outs*. Moreover, Obendorf et al. [40] used scenarios to connect a *design vision* with the more technical tasks of programmers in a *Scenario-based usability engineering* approach.

Story Cards present additional information related to user stories. Abdullah et al. [37] report that story cards allow capturing plans (estimates), history (who worked on the card) and goals. Besides, Farid [62] defined exactly which questions had to be answered with a story card in the W^8 *User Story Card Model*. The

eight "W" are: who, what, why, without ignoring, while it is nice to have, within, with a priority of and which may impact. On the contrary, Ramesh et al. [39] used story cards to document requirements for the next iteration that were elicited through communications with the customer.

Artifacts that are mentioned one time: Wall, pin board, Event-driven Process Chain (EPC) models, business process repository, domain models, snapshots, tags, SUI model, index cards, data flow diagrams, user wish list, user journey, UX requirements, design concept, evaluation goals, test specifications, role model, task model, operational model, interaction scenarios, user performance, experience goals, hedonic quality goals, document with FR and NFR, realization concept, refined into requirements lists, working software, system model, user model, effect maps, sketches, product backlog, user-diaries with picture and videos, sprint goal, delivery roadmap, definition of "done" and sprint burn-down chart.

Concerning classical agile artifacts (e.g. product backlog, sprint goal and sprint backlog) we can point out that the included studies rarely mentioned them. This leads to the conclusion that not every used artifact is reported by the studies.

4.4.2. Documentation of requirements

The results of the literature review show that there are also some problems regarding documentation of requirements. Blomkvist et al. [56] identified that not every artifact was used by developers. They report that developers did not read *personas, scenarios* and *effect maps*. They describe this phenomenon as *TAGRI principle* (They Ain't Gonna Read It). For this purpose, it is important to find the right combination of artifacts that fit the context of the project and people working in it.

Furthermore, Blomkvist et al. [56] recommend that UCD specialists should translate their work directly into user stories, otherwise user stories have a strong technical focus. They state that user stories provide a way to translate UCD work into a format that had been already used in ASD. In contrast, Cajander et al. [35] report that they are not well suited to usability work as there are difficulties to describe usability aspects in such a way. To their mind, usability needs to be addressed on a higher level.

We also find another aspect concerning the treatment of user stories. Liskin et al. [45] studied the granularity of user stories, specifically the level of functionality an artifact deals with. They verified that there were communication and planning issues for big stories (implementation < 1 week) but they were too vague, thus, they recommend splitting such stories. Furthermore, they report that requirements artifacts could avoid miscommunication and make requirements visible.

4.4.3. Functional and non-functional requirements

Several studies classify requirements as functional or non-functional (see C3.3, Table 7). However, we have uncovered some problems concerning the treatment of Non-Functional Requirements (NFR).

Ramesh et al. [39] identified, in a multi-case study, neglected non-functional requirements as a challenge of Agile RE. They report that some organizations had not been paid much attention to NFR in early development cycles and that this lack of attention has often led to redevelopment and bottlenecks.

This challenge is addressed by several studies (e.g. [52,62,58]). Bourimi et al. [52] consider NFR in early stages of the development process with their *Agile Framework for integrating non-functional Requirements Engineering (AFFINE framework).* Therefore, they introduce the role of *NRF stakeholder* into Scrum, who is responsible for managing NRFs and acts like a facilitator to all stakeholder of the project. On the contrary, Farid [62] developed an agile methodology for identifying, linking, and modeling NFRs with FRs through different kind of cases and aspect-oriented pointcut operators. In the *Non-functional Requirements Modeling for Agile Processes (NORMAP)* requirements are classified as functional or non-functional by taxonomy. Nawrocki et al. [58] looked at the elicitation of NFRs and proposed a method called *SENoR (Structured Elicitation of Non-functional Requirements)*, which consists of three steps: 1) presentation of the business case 2) series of short brainstorming session according to ISO25010 [63] and 3) voting with regard to the importance of the elicit requirements.

Another problem is reported by Lucia et al. [54]. They claim that there is often a lack of formal acceptance tests for NFRs. A similar observation was made by Dragicevic et al. [44], who recommend specifying at least one KPI to measure each NRF.

5. Discussion

In sum, we have found 27 relevant studies analyzed according to our research protocol. Next, we will discuss on the findings of our SLR. First, we will refer to the meaning of findings related to our RQs and, secondly we will identify the limitations of our study.

5.1. Meaning of findings

5.1.1. General findings

The results of the SLR show that Agile RE has been studied within various research areas (e.g. software engineering, human factors or Participatory Design). We can conclude that this is an important research topic. Furthermore, the heterogeneity of the studied aspects in the reviewed studies shows that this is a complex research field with a lot of different cross-functional influences. Moreover, we can state that this research field is very close to existing work practices in companies, since most of the included studies analyze the aspects in a real life context.

5.1.2. Findings related to RQ1

Concerning our first RQ, we can conclude that continuous communication and collaboration is the most frequent used approach to involve stakeholder in the process of RE compatible with ASD. The variety of aspects and methods reported by the reviewed studies point out that there is no common process model for stakeholder involvement in an agile environment. Although a broad range of reviewed studies dealt with the development of systems including a user interface, only half of them involved users directly into the development process. However, it is necessary to engage user in order to validate assumptions during system development; otherwise requirements would be seen as a single point of truth. In light of this, we consider that further research with regard to a structured process model for stakeholder and user involvement is required.

5.1.3. Findings related to RQ2

Studies that address the user perspective in ASD provide some insights about which methodologies are useful in order to make ASD more user-centric. We find that *Human-Centered Design, Design Thinking, Contextual Inquiry* and *Participatory Design* are commonly used methodologies. However, the results also show that there are problems with sharing knowledge between stakeholder concerning user requirements and the responsibility for usability/UX. To this end, we can conclude that it is a key to find appropriate methodologies that help to solve these problems.

5.1.4. Findings related to RQ3

The analysis of the reviewed studies shows that a variety of different artifacts are applied to Agile RE. We have identified user stories, prototypes, uses cases, scenarios and story cards as the most frequent used artifacts. Furthermore, we have faced some problems regarding requirements documentation; there are

difficulties to identify the right kind of artifacts, which enhance collaboration among stakeholder, developer and agile team. This is a special challenge with regard to the project setting (e.g. co-located or distributed teams). Nevertheless, it is important to create appropriate guidelines for requirements management within ASD.

5.2. Limitations of the review

There may be some relevant papers we missed because of the high amount of published literature, even though we used a predefined protocol and followed a rigorous search strategy to ensure the completeness of our study. We addressed this risk through forward and backward snowballing, since it also preserves us for a bias in the selection process. The selection process was mainly performed by the first author of the paper (PhD student). We applied our search strategy, due to the high amount of findings from our search. For the phases P4-P6 (scan title, abstract and content manually) the first author decided which papers were relevant to be included. This may lead to a certain degree of subjectivity while performing such a selection. However, in cases of difficult decisions, the first author consulted the others in order to reduce subjectivity.

Another possible weakness of our approach might be the chosen selection criteria. For example, we focused on papers written in English language. Therefore, there might be relevant studies written in languages other than English, which were excluded because of the applied exclusion criteria.

With regard to the limitations in data extraction, we are aware of the fact that some aspects in reviewed studies (e.g. artifacts and methods) might be poorly reported. For this purpose, our results would have been different, if the studies had been reported more accurately. We tried to address this issue through an extensive quality assessment of the included studies.

6. Conclusions and future work

This paper presents a SLR on Agile RE addressed to stakeholder and user involvement with the aim to capture the current state of the art of the literature related to the integrated field of ASD+RE+UCD. This review was conducted by following appropriate guidelines provided by Kitchenham and Charters [32]. We identified 42,808 papers in our initial search, and 965 further studies through the snowballing technique. Our search process was carried out in different phases in order to reduce the findings. In total, 27 studies were identified as relevant and analyzed. Firstly, we evaluated each paper with a quality assessment. Then, the findings were quantitatively classified according to a publication channel and research method. The included studies were published in between 2007 and 2015.

This review has several implications for both researchers and practitioners. Based on a qualitative analysis of the included studies, we can conclude that building a *shared understanding* of the *user perspective* is not very well established in ASD. It became obvious during the deeper analysis of the identified publications, that only a limited number of papers investigated the presence of the user perspective in ASD. These publications revealed that there were many problems concerning the direct involvement of users and stakeholders. However, we identified four methodologies (*Human-Centered Design, Design Thinking, Contextual Inquiry* and *Participatory Design*) that were integrated in ASD with the aim to increase the knowledge about user needs. Furthermore, we identified a broad range of different methods that can be used in ASD to *gather data* in terms of RE. We identified the following key *artifacts* for the *documentation* of requirements that are used in Agile RE: *User stories, prototypes, use cases, scenarios* and *story cards*. Industrial practitioners can utilize these findings as recommendations to discover the right combination of artifacts for their development process. With regards to *NFRs,* we can conclude that on one hand, there are different approaches to deal with NFRs, but on the other hand, we determine an open challenge concerning the estimation and measurement of these requirements (e.g. *UX metrics*, *security policies*).

To summarize, it must be stated that the review shows the need for more empirical studies that work on Agile RE using various kinds of project settings (e.g. different agile methodologies, scaling or distance of project member). In addition, it can be concluded that there is heterogeneity among Agile RE approaches focusing on user and stakeholder involvement. Future research may specifically deal with a commonly used process model for stakeholder and user involvement in ASD. To this end, we will create a metamodel that describes the influencing parameters on Agile RE at a higher level. Furthermore, appropriate methodologies have to be found for building a shared understanding concerning the user perspective among project members and stakeholders. Our review of Agile RE studies also shows that the topic of requirements management needs further empirical evaluation due to the lack of appropriate guidelines in practice.

Acknowledgements

This research has been supported by the Megus project (TIN2013-46928-C3-3-R) and by the SoftPLM Network (TIN2015-71938-REDT) of the Spanish Ministry of Economy and Competitiveness.

Appendix A

[P1] N.N.B. Abdullah, S. Honiden, H. Sharp, B. Nuseibeh, D. Notkin, *Communication patterns of agile requirements engineering*, in: Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11, ACM Press, New York, New York, USA, 2011: pp. 1–4.

[P2] S. Adikari, C. McDonald, J. Campbell, *Reframed contexts: Design thinking for agile user experience design*, in: Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013.

[P3] A. Bellucci, G. Jacucci, V. Kotkavuori, B. Serim, I. Ahmed, S. Ylirisku, *Extreme Co-design: Prototyping with and by the User for Appropriation of Web-connected Tags*, in: 2015: pp. 109–124.

[P4] J.K. Blomkvist, J. Persson, J. Åberg, *Communication through Boundary Objects in Distributed Agile Teams*, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15, ACM Press, New York, New York, USA, 2015: pp. 1875–1884.

[P5] M. Bourimi, T. Barth, J.M. Haake, B. Ueberschär, D. Kesdogan, *AFFINE for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies*, Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 6409 LNCS (2010) 182–189.

[P6] J. Buchan, *An Empirical Cognitive Model of the Development of Shared Understanding of Requirements*, in: Communications in Computer and Information Science, 2014: pp. 165–179.

[P7] Å. Cajander, M. Larusdottir, J. Gulliksen, *Existing but Not Explicit - The User Perspective in Scrum Projects in Practice*, in: Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013: pp. 762–779.

[P8] S. Dragicevic, S. Celar, L. Novak, *Use of Method for Elicitation, Documentation, and Validation of Software User Requirements*

(MEDoV) in Agile Software Development Projects, in: 2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks, IEEE, 2014: pp. 65–70.

[P9] W.M. Farid, *The NORMAP Methodology: Lightweight Engineering of Non-functional Requirements for Agile Processes*, in: 2012 19th Asia-Pacific Software Engineering Conference, IEEE, 2012: pp. 322–325.

[P10] M. Harbers, C. Detweiler, M.A. Neerincx, *Embedding Stakeholder Values in the Requirements Engineering Process*, Springer International Publishing, Cham, 2015.

[P11] A.A. Issa, A.I. AlAli, *Automated requirements engineering: use case patterns-driven approach*, IET Software. 5 (2011) 287.

[P12] P. Kamthan, *On the interaction between socio-technical dimensions in the next generation of requirements engineering*, International Journal of Software Engineering and Its Applications. 7 (2013) 183–196.

[P13] K. Kautz, *Participatory Design Activities and Agile Software Development*, in: Proceedings of the IFIP WG 8.2/8.6 International Working Conference, Perth, Australia, 2010: pp. 303–316.

[P14] S.-H. Lee, I.-Y. Ko, S. Kang, D.-H. Lee, *A Usability-Pattern-Based Requirements-Analysis Method to Bridge the Gap between User Tasks and Application Features*, in: 2010 IEEE 34th Annual Computer Software and Applications Conference, IEEE, 2010: pp. 317–326.

[P15] O. Liskin, K. Schneider, F. Fagerholm, J. Münch, *Understanding the role of requirements artifacts in kanban*, in: Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014, ACM Press, New York, New York, USA, 2014: pp. 56–63.

[P16] B. Losada, M. Urretavizcaya, I. Fernández-Castro, *A guide to agile development of interactive software with a "User Objectives"-driven methodology*, Science of Computer Programming. 78 (2013) 2268–2281.

[P17] A. De Lucia, A. Qusef, *Requirements Engineering in Agile Software Development*, Journal of Emerging Technologies in Web Intelligence. 2 (2010) 212–220.

[P18] M. Maguire, *Using human factors standards to support user experience and agile design*, in: Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013.

[P19] I. Mahmud, V. Veneziano, *Mind-mapping: An effective technique to facilitate requirements engineering in agile software development*, in: 14th International Conference on Computer and Information Technology (ICCIT 2011), IEEE, 2011: pp. 157–162.

[P20] T. Memmel, F. Gundelsweiler, H. Reiterer, *Agile Human-Centered Software Engineering*, Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI…but Not as We Know It - Volume 1. (2007) 167–175.

[P21] P. Näkki, K. Koskela, M. Pikkarainen, *Practical model for user-driven innovation in agile software development*, in: Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE 2011), IEEE, 2011: pp. 1–8.

[P22] J. Nawrocki, M. Ochodek, J. Jurkiewicz, S. Kopczyńska, B. Alchimowicz, *Agile requirements engineering: A research perspective*, in: Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014: pp. 40–51.

[P23] H. Obendorf, M. Finck, *Scenario-based usability engineering techniques in agile development processes*, in: Proceeding of the Twenty-Sixth Annual CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI '08, ACM Press, New York, New York, USA, 2008: p. 2159.

[P24] H.H. Olsson, J. Bosch, *Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation*, LNBIP. 210 (2015) 154–166.

[P25] B. Ramesh, L. Cao, R. Baskerville, *Agile requirements engineering practices and challenges: an empirical study*, Information Systems Journal. 20 (2010) 449–480.

[P26] J.M. Rivero, J. Grigera, G. Rossi, E. Robles Luna, F. Montero, M. Gaedke, *Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering*, Information and Software Technology. 56 (2014) 670–687.

[P27] F. Wanderley, A. Silva, J. Araujo, D.S. Silveira, *SnapMind: A framework to support consistency and validation of model-based requirements in agile development*, in: 2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE), IEEE, 2014: pp. 47–56.

References

[1] T. Dingsøyr, T. Dyba, Empirical studies of agile software development: A systematic review, Information and Software Technology 50 (2008) 833–859.

[2] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, et al., Manifesto for Agile Software Development, (2001). ⟨http://www.agilemanifesto.org/⟩(accessed July 16, 2016).

[3] K. Schwaber, Agile Project Management with Scrum, Microsoft, 2004.

[4] D.J. Anderson, Kanban - Successful Evolutionary Change for your Technology Business, Blue Hole Press, 2010.

[5] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley, 2000.

[6] International Organization for Standardization, ISO 9241-210:2010 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems, 2010.

[7] T. Pfeiffer, J. Hellmers, E.-M. Schön, J. Thomaschewski, Empowering User Interfaces for Industrie 4.0, Proceedings of the IEEE. 104 (2016) 986–996.

[8] F. Paetsch, a. Eberlein, F. Maurer, Requirements engineering and agile software development, WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. (2003) 1–6.

[9] G. Sillitti, Alberto and Succi, Requirements engineering for agile methods, Engineering and Managing Software Requirements (2005) 309–326.

[10] I. Sommerville, P. Sawyer, Requirements Engineering: A Good Practice Guide, 1st ed, John Wiley & Sons, Inc, New York, USA, 1997.

[11] K. Pohl, Requirements Engineering: Fundamentals, Principles, and Techniques, 1st ed, Springer Publishing Company, Incorporated, 2010.

[12] S. Adikari, C. McDonald, J. Campbell, Little design up-front: A design science approach to integrating usability into agile requirements engineering, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2009.

[13] H. Takeuchi, I. Nonaka, H. Takeuchi, The new new product development game, Harvard Business Review 64 (1986) 137–146.

[14] P. Kruchten, The Rational Unified Process, 1st Editio, Addison-Wesley, 1998.

[15] P. Kruchten, The Rational Unified Process: An Introduction, 3rd ed, Addison-Wesley, 2004.

[16] K. Schwaber, SCRUM Development Proces, in: J. Sutherland, C. Casanave, J. Miller, P. Patel, G. Hollowell (Eds.), s, Business Object Design and Implementation, London, 1997.

[17] S.R. Palmer, M. Felsing, A Practical Guide to Feature-Driven Development, Pearson Education, 2001.

[18] D.J. Anderson, Making the Business Case for Agile Management - Simplifying the Complex System of Software Engineering, in: Motorola S3 Symposium, 2004: pp. 1–13.

[19] E.M. Schön, M. Escalona, J. Thomaschewski, Agile Values and Their Implementation in Practice, International Journal of Interactive Multimedia and Artificial Intelligence 3 (2015) 61.

[20] M. Cohn, User Stories Applied: For Agile Software Development, 2004.

[21] J. Patton, User Story Mapping: Discover the Whole Story, Build the Right Product, First edit, O´Reilly, 2014.

[22] W. Hudson, User stories don't help users, Interactions 20 (2013) 50–53.

[23] E. Schön, D. Winter, J. Uhlenbrok, M.J. Escalona, Enterprise Experience into the Integration of Human-Centered Design and Kanban, in: Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016), Lisbon, Portugal, 2016: pp. 133–140.

[24] J. Humble, D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, Inc, Boston, MA, 2010.

[25] M.K. Larusdottir, E.R. Bjarnadottir, J. Gulliksen, The Focus on Usability in Testing Practices in Industry, in: Proceedings of Second IFIP TC 13 Symposium, HCIS 2010, Held as Part of WCC 2010, 2010: pp. 98–109.

[26] E. Bjarnason, M. Unterkalmsteiner, M. Borg, E. Engstr??m, Multi-case study of agile requirements engineering and the use of test cases as requirements, Information and Software Technology 77 (2015) 61–79.

[27] T. Silva da Silva, A. Martin, F. Maurer, M. Silveira, User-Centered Design and Agile Methods: A Systematic Review, in: 2011 AGILE Conference, IEEE, 2011: pp. 77–86.

[28] D. Salah, R.F. Paige, P. Cairns, A Systematic Literature Review for Agile Development Processes and User Centred Design Integration, in: Proceedings of the

18th International Conference on Evaluation and Assessment in Software Engineering, 2014: pp. 5:1–5:10.
[29] M. Brhel, H. Meth, A. Maedche, K. Werder, Exploring principles of user-centered agile software development: A literature review, Information and Software Technology 61 (2015) 163–181.
[30] I. Inayat, S.S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, Computers in Human Behavior 51 (2015) 915–929.
[31] H.F. Soares, N.S.R. Alves, T.S. Mendes, M. Mendonca, R.O. Spinola, Investigating the Link between User Stories and Documentation Debt on Software Projects, in: 2015 12th International Conference on Information Technology - New Generations, IEEE, 2015: pp. 385–390.
[32] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, Engineering 2 (2007) 1051.
[33] E.-M. Schön, J. Thomaschewski, M.J. Escalona, Agile Requirements Engineering - Protocol for a Systematic Literature Review, (2016).
[34] S. Jalali, C. Wohlin, Systematic literature studies: database searches vs. backward snowballing, in: Proceedings of the 2012 6th ACM_IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2012: pp. 29–38.
[35] Å. Cajander, M. Larusdottir, J. Gulliksen, Existing but Not Explicit - The User Perspective in Scrum Projects in Practice, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013: pp. 762–779.
[36] B. Losada, M. Urretavizcaya, I. Fernández-Castro, A guide to agile development of interactive software with a "User Objectives"-driven methodology, Science of Computer Programming 78 (2013) 2268–2281, http://dx.doi.org/10.1016/j.scico.2012.07.022.
[37] N.N.B. Abdullah, S. Honiden, H. Sharp, B. Nuseibeh, D. Notkin, Communication patterns of agile requirements engineering, in: Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11, ACM Press, New York, New York, USA, 2011: pp. 1–4.
[38] K. Kautz, Participatory Design Activities and Agile Software Development, in: IFIP WG 8.2/8.6 International Working Conference, 2010: pp. 303–316.
[39] B. Ramesh, L. Cao, R. Baskerville, Agile requirements engineering practices and challenges: an empirical study, Information Systems Journal 20 (2010) 449–480.
[40] H. Obendorf, M. Finck, Scenario-based usability engineering techniques in agile development processes, in: Proceeding of the Twenty-Sixth Annual CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI '08, ACM Press, New York, New York, USA, 2008: p. 2159.
[41] A. Bellucci, G. Jacucci, V. Kotkavuori, B. Serim, I. Ahmed, S. Ylirisku, Extreme Co-design: Prototyping with and by the User for Appropriation of Web-connected Tags, in: 2015: pp. 109–124.
[42] M. Harbers, C. Detweiler, M.A. Neerincx, Requirements Engineering: Foundation for Software Quality, Springer International Publishing, Cham, 2015.
[43] H.H. Olsson, J. Bosch, Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation, LNBIP 210 (2015) 154–166.
[44] S. Dragicevic, S. Celar, L. Novak, Use of Method for Elicitation, Documentation, and Validation of Software User Requirements (MEDoV) in Agile Software Development Projects, in: 2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks, IEEE, 2014: pp. 65–70.
[45] O. Liskin, K. Schneider, F. Fagerholm, J. Münch, Understanding the role of requirements artifacts in kanban, in: Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014, ACM Press, New York, New York, USA, 2014: pp. 56–63.
[46] J.M. Rivero, J. Grigera, G. Rossi, E. Robles Luna, F. Montero, M. Gaedke, Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering, Information and Software Technology 56 (2014) 670–687.
[47] F. Wanderley, A. Silva, J. Araujo, D.S. Silveira, SnapMind: A framework to support consistency and validation of model-based requirements in agile development, in: 2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE), IEEE, 2014: pp. 47–56.
[48] P. Kamthan, On the interaction between socio-technical dimensions in the next generation of requirements engineering, International Journal of Software Engineering and Its Applications 7 (2013) 183–196.
[49] M. Maguire, Using human factors standards to support user experience and agile design, in: Proceedings - International Conference, UAHCI 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, 2013.
[50] P. Näkki, K. Koskela, M. Pikkarainen, Practical model for user-driven innovation in agile software development, in: Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE 2011), IEEE, 2011: pp. 1–8.
[51] I. Mahmud, V. Veneziano, Mind-mapping: An effective technique to facilitate requirements engineering in agile software development, in: 14th International Conference on Computer and Information Technology (ICCIT 2011), IEEE, 2011: pp. 157–162.
[52] M. Bourimi, T. Barth, J.M. Haake, B. Ueberschär, D. Kesdogan, AFFINE for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 6409 LNCS (2010) 182–189.
[53] S.-H. Lee, I.-Y. Ko, S. Kang, D.-H. Lee, A Usability-Pattern-Based Requirements-Analysis Method to Bridge the Gap between User Tasks and Application Features, in: 2010 IEEE 34th Annual Computer Software and Applications Conference, IEEE, 2010: pp. 317–326.
[54] A. De Lucia, A. Qusef, Requirements Engineering in Agile Software Development, Journal of Emerging Technologies in Web Intelligence 2 (2010) 212–220.
[55] T. Memmel, F. Gundelsweiler, H. Reiterer, Agile Human-Centered Software Engineering, Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but Not as We Know It - Volume 1. (2007) 167–175.
[56] J.K. Blomkvist, J. Persson, J. Åberg, Communication through Boundary Objects in Distributed Agile Teams, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15, ACM Press, New York, New York, USA, 2015: pp. 1875–1884.
[57] J. Buchan, An Empirical Cognitive Model of the Development of Shared Understanding of Requirements, in: Communications in Computer and Information Science, 2014: pp. 165–179.
[58] J. Nawrocki, M. Ochodek, J. Jurkiewicz, S. Kopczyńska, B. Alchimowicz, Agile requirements engineering: A research perspective, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014: pp. 40–51.
[59] A.A. Issa, A.I. AlAli, Automated requirements engineering: use case patterns-driven approach, IET Software 5 (2011) 287.
[60] S. Adikari, C. McDonald, J. Campbell, Reframed contexts: Design thinking for agile user experience design, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013.
[61] H. Beyer, K. Holtzblatt, Contextual design defining customer-centered systems, Morgan Kaufmann, San Francisco, Calif, 1998.
[62] W.M. Farid, The NORMAP Methodology: Lightweight Engineering of Non-functional Requirements for Agile Processes, in: 2012 19th Asia-Pacific Software Engineering Conference, IEEE, 2012: pp. 322–325.
[63] ISO, ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, 2011.