Computer Standards & Interfaces 64 (2019) 61–77

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

Usability in agile software development: A tertiary study

Karina Curcio, Rodolfo Santana, Sheila Reinehr, Andreia Malucelli*

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

ARTICLE INFO

Keywords:
Agile software development
Usability
User-centered design
Systematic literature review
Tertiary study

ABSTRACT

In the last years the interest in developing research on integration of usability and agile software development has been increasing. The number of systematic literature reviews, systematic mapping studies and non-systematic reviews, related to this thematic has also increased. Nevertheless, there is no analysis on the quality of these published secondary studies, nor is there a consolidated research that brings the answer of how to integrate these two areas. The goal of this paper is to categorize secondary studies related to the integration of usability and agile software development and present a critical analysis on the quality of the selected studies. To accomplish this goal a tertiary study was performed to categorize the related studies selected. Initially 3,065 papers were identified and further narrowed to 14 by applying exclusion criteria and analysis. We classified the selected studies as systematic literature reviews, systematic mapping studies and non-systematic literature reviews to report the data analysis. As a result of this study different forms to integrate usability and agile software development were detected as well as the various challenges that must be overcome for the integration success. Six main categories were identified to represent ways of integrating usability into agile development: processes, techniques, practices, recommendations, principles and different approaches. Regarding to the challenges for the integration seven main categories were also identified: issues related to tests, time, work balance, modularization, feedback, prioritization, and documentation. Although the interest in researching the integration of usability and agile software development has increased in the last years, mostly of the analyzed studies neglected the quality criteria and presented difficulties to use methods to synthetize the research results. Despite this, it has been realized that the integration of usability with agile software development is possible and is strongly aligned with user-centered design. The initial studies indicated a separation of activities and roles into specific tracks with parallel work to treat usability in agile software development, but the trend is no longer to manage and control these activities in separate ways, so new challenges are becoming to appear. Although we have identified several points of tension, the integration does not become unfeasible.

1. Introduction

In the last years users have become more experienced in the use of software systems and much more demanding for quality products. For this scenario agile methodologies speed up the process of building software with more frequent deliveries to customers. Nevertheless, agile practitioners often drive their attention to understand customer's needs, trying to answer the question of how useful the developed software can be, focusing on providing an appropriate functional scope. They often do not focus on developing software that is considered usable by the customer [1–3]. During the agile development customers are usually closer and more involved with the process making it easier to provide constant feedback. These feedbacks are valuable to have a good user experience. The problem is that the agile practitioners usually focus on delivering the working software but neglect the user experience, and hardly discuss about users or users' interface [4]. In some agile methodologies or frameworks like Scrum, Extreme Programming (XP), and Feature-driven Development (FDD) there is no defined role that represents a user interface specialist or an interaction designer. This role in agile teams is not clear and largely overlooked [S6]. Moreover, none of the major agile methods include guidance for the practitioner to develop usable software [5]. As described by Salah, Paige and Cairns [S6], "practices for understanding, eliciting usability and user requirement and evaluating agile systems for usability and user experience are generally considered deficient". In software engineering the usability is not a central topic because it is considered one of many non-functional requirements and quality attributes [6], but it has become recognized as a crucial point for the success in highly competitive markets [S9]. In this context the need for a better integration between usability and the agile software development process

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

https://doi.org/10.1016/j.csi.2018.12.003
Received 16 July 2018; Received in revised form 31 December 2018; Accepted 31 December 2018
Available online 03 January 2019
0920-5489/

arises.

During our initial investigation, exploring this thematic, we did not find any study that could bring us a consolidated view about the available literature review related to this possible integration. For this reason, we decided to develop a tertiary study that aimed to identify not only systematic literature reviews (SLR) but also other types of secondary studies, including systematic mapping studies and non-systematic literature reviews to enrich the research, discussion, and also the results. Our goal was to identify how the recent studies are trying to integrate these two areas, analyze and discuss the quality of these studies, and map what are the challenges in the integration between these two areas of knowledge. Through answering the research questions and topics presented in this study we intended to provide knowledge about the addressed topic for future works.

This work is structured as follows. Section 2 presents the main concepts of usability and agile software development. Section 3 explains the method and the search strategies that were used in this article. Section 4 presents the primary results found and also the answers of the research questions. Section 5 presents a discussion of the results. In Section 6 the threats to the validity of the study are described. Finally, in Section 7 the conclusion of the research is presented.

2. Background

2.1. Usability

The term “usability” is related to software products quality studies as well as to ergonomics, interaction design, and human-computer interaction studies. Usability is defined by the International Organization for Standardization ISO 9241-11:1998 [7] as "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Usability is also a sub-characteristic of software quality previously defined by ISO 9126 [8] and substituted by the ISO 25000 family. The software quality model ISO/IEC 25010:2011 [9] describes this as "quality in use", to distinguish it from narrower interpretations of usability as just the ease of use of an interface. The term usability is also associated with other terms, sometimes erroneously treated as synonyms, such as usability engineering, user experience (UX), interaction design (IxD) and user-centered design (UCD).

Usability engineering was defined by Tyldesley [10] as “a process whereby the usability of a product is specified quantitatively, and in advance. Then, as the product is built, testing takes place to see whether the planned-for levels of usability have been achieved”. The aim of usability engineering is to engineer for improvement and its nature is reflected in an iterative development with cycles of design-evaluate-redesign [11]. The usability engineering set five main usability characteristics that are related to issues such as system learnability, efficiency, memorability, errors and user satisfaction [12].

User experience is defined by ISO 9241-210:2010 [13] as “person's perceptions and responses resulting from the use and or anticipated use of a product, system or service”. So, the user experience is subjective, and its focus is on use. While usability is generally considered the user's ability to use something to complete a task successfully, the user experience has a broader view, looking at the individual's complete interaction, thoughts, feelings and perceptions that result of this interaction [14].

The definition of interaction design by Sharp, Rogers and Preece [15] is "designing interactive products to support people in their everyday and working lives". It means creating experiences that enhance and extend the way people work, communicate, and interact, regardless of the type of product. Interaction design is concerned with a broader scope of issues, topics and paradigms including areas as psychology, human-computer interaction, web design, computer science, information systems, marketing, entertainment and business. Therefore, usability and user experience are issues that must be addressed within the design of iterative systems.

User-centered design is a design process that focuses on user needs and requirements. It was first defined by the ISO 13407:1999 [16] and includes a general process for including user-centered activities throughout a development lifecycle without specifying the exact methods to develop it. This standard is related to ergonomics catalogue and basically it defines four activities that form the main cycle of work: specify the context of use, specify the requirements, create design solutions and evaluate the design. This standard has been revised by ISO 9241-210:2010 [13].

2.2. Agile software development

In early 2001 seventeen software development professionals met in Utah to discuss and share new approaches to software development. As a result of this meeting the "Manifesto for Agile Software Development" [17] was created. The manifesto consisted of a group of values and principles, which aimed to help people to understand the concept of the agile software development. The values stated were: individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and, responding to change over following a plan. Several methods and frameworks for agile software development emerged based on the values and principles established there, although the Manifesto itself was a compilation of already existing agile methodologies. In their article Dingsøyr et al. [18] synthesizes the main methods of agile software development: Crystal methodologies [19], Dynamic Software Development Method (DSDM) [20], Feature-driven Development [21], Lean Software Development [22], SCRUM [23] and eXtreme Programming (XP) [24]. As described by Dingsøyr et al. [25] agile methods "are iterative, with focus on teamwork, client-developer collaboration, customer feedback throughout the software project lifecycle, and support for advance product delivery". The Agile Manifesto emphasized in giving more value to the customer collaboration, individuals and interactions producing a minimum documentation and incremental features. With this way of working is intended to have early and continuous delivery to the customer and receive a continuous feedback. This has been identified as a successful process, as the frequently deliveries make it easier to get the customer closer and involved from the beginning of the project. However, the focus on delivering functionalities can have some usability impacts as the agile methods usually have short iterations with minimal up-front design and focused on functional tests.

3. Research method

This tertiary study was developed following the guidelines proposed by Kitchenham and Charters [26] for performing systematic literature reviews in software engineering. The guidelines were created to assist researchers, meta-analysts and reviewers in designing, conducting, and evaluating empirical research. To conduct the research specific phases are proposed: planning, conducting and reporting the review. The planning phase is dedicated to identifying the need for the review, including goals and research questions, and also the search strategy, including the search string and the inclusion/exclusion criteria, as detailed in the next subsections.

3.1. Objectives and research questions

A tertiary study aims to provide information about the available literature reviews on a specific topic and tabulate information like the number of publications, their quality, and the focus of those publications within the topic of investigation.

The main objective of this study was mapping the information provided by secondary studies on the integration of agile development methodologies and the concern with usability. For this it was intended

to answer the following research questions:

1) What research questions were investigated in the secondary studies?
2) What are the main ways to integrate usability and agile software development according to the secondary studies?
3) What are the indicators of the quality of the secondary studies?
4) What challenges are described in the published studies related to the integration of usability and agile software development?

3.2. 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 February of 2018. This date was defined because 2001 was the year of the Agile Manifesto release;
2) Studies related to the search string defined;
3) Peer reviewed studies;
4) Secondary studies.

The exclusion criteria were:

1) The study is not labeled as a secondary study;
2) The study is a secondary study, but the subject was not directly related to integration of usability and agile software development;
3) Duplicated papers.

The search string defined was ((agile OR xp OR "extreme programming" OR scrum) AND (usability OR "user-centered design" OR UCD OR "user interaction" OR "user experience") AND ("systematic literature review" OR SLR OR "systematic review" OR "systematic mapping" OR "mapping study" OR "literature review")). The search was executed in February 2018 and returned 3065 papers. The steps for conducting the review are shown in Fig. 1.

A peer review strategy was adopted, and the first two authors analyzed all returned studies. Each of the reviewers received a spreadsheet with the search string result from each of the selected databases. All papers were therefore analyzed and classified according to the pre-established inclusion and exclusion criteria. At the end of this stage the results of each member were analyzed and compared. Discussions were made to reach consensus for every conflicted result. At the end a total of 14 papers were then selected as a result of the classification, while 3051 papers were excluded, as demonstrated in Table 1.

3.3. Quality assessment

In this tertiary study a quality assessment was used to evaluate each publication using the same set of criteria adopted by Kitchenham et al. in their tertiary studies [28,29]. These quality assessment criteria were defined by the Centre for Reviews and Disseminations (CRD) Database of Abstracts of Reviews of Effects (DARE), of the York University [30]. The criteria are based on five quality assessment questions and were scored as described in Table 2.

The quality assessment was made by two authors and a final score was extracted. After the assessment of all selected studies we decided not to exclude the ones with low scores. We took advantage of the results to provide more data for our analysis and discussion. So, a total amount of 14 (shown in Appendix A) studies remained for the analysis and extraction of the results.

3.4. Data extraction

For all of these selected studies, the following data were extracted:

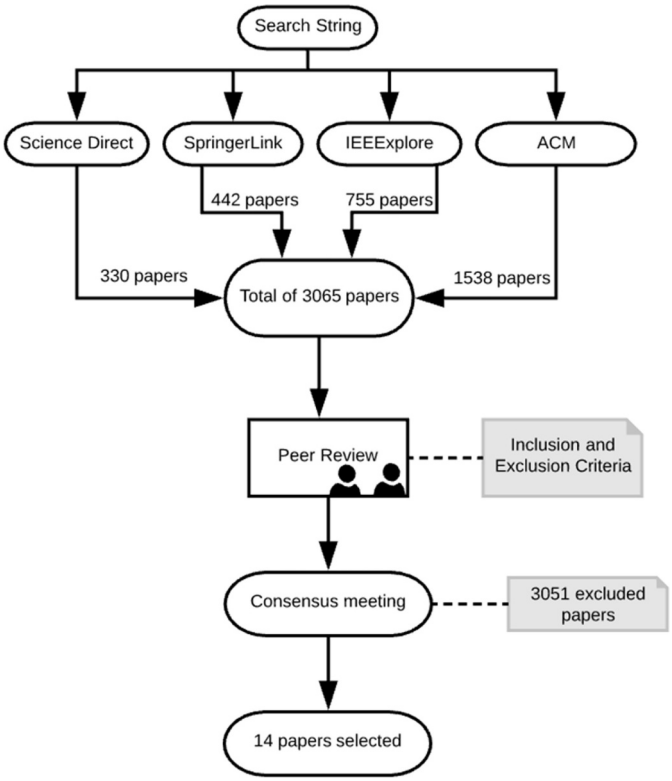

Fig. 1. Graphical representation of the steps for the selection of papers.

Table 1
Number of papers selected during the systematic mapping study.

Databases	Total of identified papers	Number of papers	
		Included	Excluded
ScienceDirect	330	0	330
SpringerLink	442	3	439
IEEExplore	755	6	749
ACM	1538	5	1533
Total	3065	14	3051

title, authors, country/location, database, publication source, year of publication, type of research (systematic literature review, systematic mapping study, non-systematic literature review), goal of research, research questions, quality assessment score and limitations.

For the classification of the studies regarding to the type of research, it was analyzed whether the studies followed some type of research protocol for systematization, such as systematic literature review by Kitchenham and Charters [26] or systematic mapping studies by Petersen et al. [27] and its depth. We also considered the classification, previously presented by the authors, indicating the type of research developed with the respective justifications.

4. Results

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

4.1. Overview of the selected papers

To demonstrate the results of this study a narrative synthesis was applied [31]. Narrative synthesis is a commonly used method to synthesize researches, including systematic reviews, with the adoption of a narrative approach (opposed to statistical) to summarize the findings of the study. According to Cruzes and Dybå [32] “it is a general framework

Table 2
Criteria used for the quality assessment.

1. Are inclusion/exclusion criteria reported related to the primary studies, which address the review question?	**Inclusion/exclusion criteria** Y (yes), the inclusion criteria are explicitly defined in the study, score 1; P (partly) the inclusion criteria are implicit, score 0.5; N (no), the inclusion criteria are not defined and cannot be readily inferred, score 0.
2. Is there evidence of a substantial effort to search for all relevant research?	**Adequacy of search** Y (yes), the authors have either searched 4 or more digital libraries and included additional search strategies or identified and referenced all journals addressing the topic of interest, score 1; P (partly), the authors have searched 3 or 4 digital libraries with no extra search strategies or they searched a defined but restricted set of journals and conference proceedings, score, 0.5; N (no), the authors have searched up to 2 digital libraries or an extremely restricted set of journals, score 0.
3. Are the primary studies summarized appropriately?	**Synthesis method** Y (yes), an explicit synthesis method is named and a reference to the method is supplied, score 1; P (partly), a synthesis method is named but no reference to the method is supplied, score 0.5 N (no), no synthesis method is named, score 0.
4 Is the validity of included studies adequately assessed?	**Quality criteria** Y (yes), the authors have explicitly defined quality criteria and extracted them from each primary study, score 1; P (partly), the research question involves quality issues that are addressed by the study, score 0.5; N (no), no explicit quality assessment of individual primary studies has been attempted or authors have defined quality criteria but not used criteria, score 0.
5. Is sufficient detail of the individual studies presented?	**Information provided about primary studies** Y (yes), information is presented about each primary study, score 1; P (partly), only summary information is presented about papers, score 0.5; N (no), the results of the individual studies are not specified, score 0.

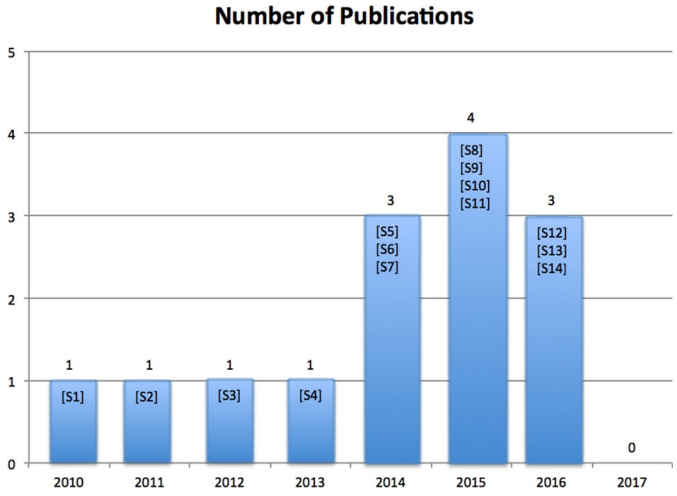

Fig. 2. Distribution of the published secondary studies reviewed per year.

of selected narrative descriptions and ordering of primary evidence with commentary and interpretation combined with specific tools and techniques that help to increase transparency and trustworthiness". To support this narrative, we classified and presented the results in a chronological order. The first chart in Fig. 2 presents the distribution of publications per year. It is important to observe that our research string was limited by the Agile Manifesto (2001) [17], but the first systematic literature review trying to integrate usability and agile software development only appeared in 2011 [S2]. The reason for this can have several sources, among them we can mention: at the time there might not have enough work in the area that would justify a systematic review, or some other term or concept may have been used at the time and so the search terms used in this study did not identify such papers, since the vocabulary used some years ago could be different.

Two interesting information that appeared after the classification of the secondary studies were related to the type of research and the related publication sources as demonstrated in Table 3. Different from the current tertiary studies that considered only systematic literature reviews, in this work we decided to include non-systematic literature reviews ([S1], [S3], [S11], [S14]) and systematic mapping studies ([S7], [S10], [S12], [S13]), as we considered the authors as being relevant references in this subject and the inclusion of these researches could enrich our analysis. As reported by Petersen et al. (2008) "there are differences with respect to the research questions, search process, search strategy requirements, quality evaluation, and results" when contrasted the different characteristics of the process of systematic literature reviews and mapping studies. He also argues that "quality assessment is more essential in systematic reviews to determine the rigor and relevance of the primary studies. In systematic maps no quality assessment needs to be performed". From this information we also took into consideration the depth of the quality control of the works for the analysis proposed in this work.

In Fig. 3 it is possible to identify the countries in which the authors that developed the selected studies are located. Overall 13 different countries were found, and some of the selected studies were developed in collaboration, so that is why some of the references appeared in more than one country. During the analysis it was possible to detect that the authors were distributed within the four continents: America (Canada, USA, Peru, Chile and Brazil), Europe (Spain, Germany, UK, Norway and Sweden), Asia (Pakistan, India) and Oceania (New Zealand).

4.2. Answering the research questions

RQ1) What research questions were investigated in the secondary studies?

Table 4 presents a summary of the selected papers including information as the database source, authors, goal, research questions, number of primary studies selected and a synthesis of the answers of each research question.

RQ2) What are the main ways to integrate usability and agile software development according to the secondary studies?

Table 5 presents the mapped ways to integrate usability and agile software development. From the secondary studies, it was identified and extracted the main forms of integration, the description of how this integration occurred, as well as, if there was any other complementary process to the agile method used to achieve this integration and the respective authors.

To make the ways of integrating usability into agile development even clearer, it was decided to group them to create generic main categories. From this, six categories were identified: processes, techniques, practices, recommendations, principles, and different approaches, as described in Table 6.

Table 3
Types of research and related publication source.

Ref.	Type of Research	Publication Source
[S1]	Non-systematic review	International Conference on Computer Design and Applications (ICCDA)
[S2]	Systematic review	Agile Conference (AGILE)
[S3]	Non-systematic review	International Journal of Agile and Extreme Software Development (IJAESD)
[S4]	Systematic review	International Conference on Information Systems Development (ISDEVEL)
[S5]	Systematic review	Euro American Conference on Telematics and Information Systems (EATIS)
[S6]	Systematic review	International Conference on Evaluation and Assessment in Software Engineering (EASE)
[S7]	Systematic mapping study	Agile Conference (AGILE)
[S8]	Systematic review	International Conference on Design, User Experience and Usability: Design Discourse (DUXU)
[S9]	Systematic review	Information and Software Technology
[S10]	Systematic mapping study	International Conference on Computational Science and its Applications (ICCSA)
[S11]	Non-systematic review	International Conference on Data and Software Engineering (ICoDSE)
[S12]	Systematic mapping study	XLII Latin American Computing Conference (CLEI)
[S13]	Systematic mapping study	IEEE International Conference on Automation (ICA-ACCA)
[S14]	Non-systematic review	International Conference on Human-Computer Interaction (HCI)

RQ3) What are the indicators of the quality of the secondary studies?

All 14 selected studies were submitted to a quality assessment made by two researchers. The scores obtained by each criterion analyzed are presented in Table 7. As described in Section 3.3 the criteria to assess the quality were defined by the Centre for Reviews and Disseminations (CRD) Database of Abstracts of Reviews of Effects (DARE), of the York University [30]. Based on five quality questions it was possible to determinate the final scores of each publication selected. Table 7 describes: (a) the type of research, which includes systematic literature reviews (SLR), systematic mapping studies (SMS) and non-systematic literature reviews (NSLR); (b) authors; (c) quality criteria (presence of inclusion/exclusion criteria clearly stated), adequacy of search, synthesis method, quality criteria, and information provided about primary studies upon which they were based; (d) their individual final score (sum of the previous grades).

The selected papers presented in Table 7 were separated according to the type of work proposed (systematic literature reviews, systematic mapping studies and non-systematic literature reviews) in a chronological order. We made this separation because Kitchenham and Charters [26] contrasted some of the differences between the process of systematic literature reviews and mapping studies. According to them there are differences in relation to the research questions, search process, search strategy requirements, quality evaluation, and in the results.

According to Petersen et al. (2008) systematic mapping studies are used to structure a research area. The research questions are broader as they aim to discover research trends. On the other hand, systematic reviews are focused on gathering and synthesizing evidence through specific research questions. Petersen et al. (2008) also affirms that "quality assessment is more essential in systematic reviews to determine the rigor and relevance of the primary studies. In systematic maps no quality assessment needs to be performed".

In contrast, non-systematic reviews usually do not follow a research protocol. They are usually used as a scoping study, which is less likely to address specific research questions or assess the quality of the selected studies. According to Cruzes and Dybå [32] "the quality of a review depends on the extent to which scientific and transparent review methods were used to minimize error and bias." As previously described, we decided to keep in the analysis non-systemic secondary studies because we believe that the authors and their contributions are important for the area.

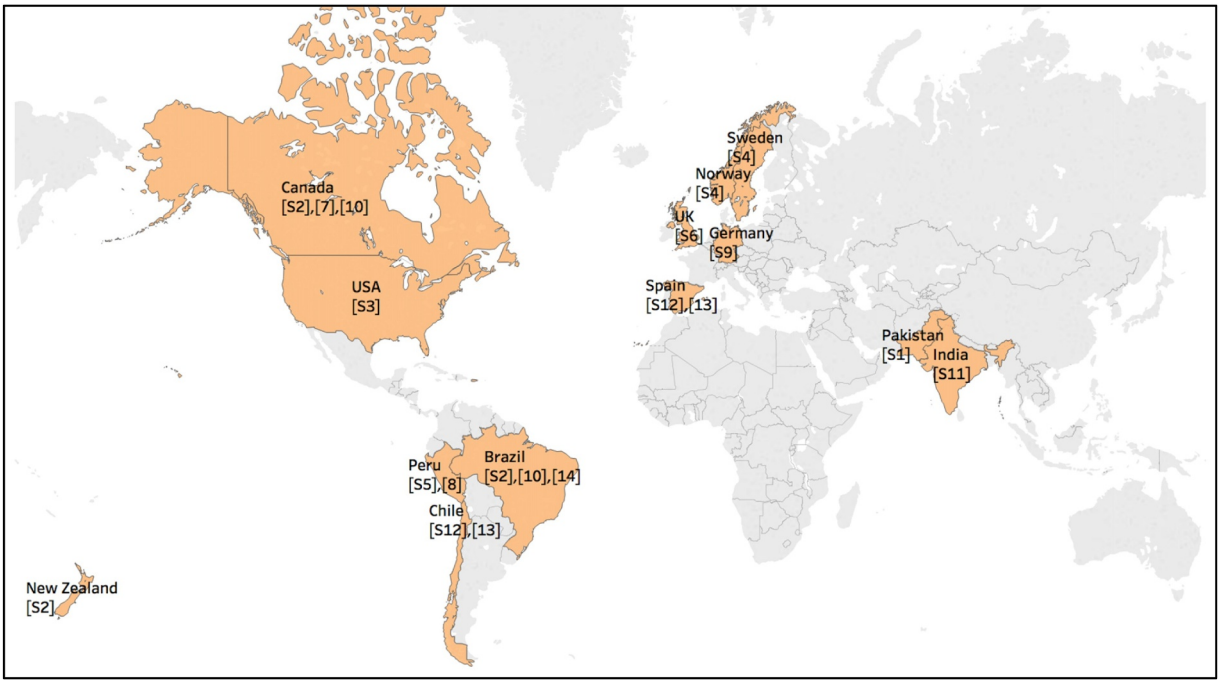

Fig. 3. Location of authors from the analyzed studies.

Table 4
Systematic reviews, mapping studies and non-systematic reviews on integrating usability in agile software development.

Database	Authors	Goal	Article research questions	Number of studies
IEEE	Sohaib and Khan [S1]	This study is a literature review that focuses on identifying the key points of tension between usability and agile methods and also on understanding how usability-engineering practices should be integrated with agile software development in order to provide effective usable software system.	(ARQ1.1) What tensions between usability engineering and agile methods have been identified in related research that makes them difficult to integrate? (ARQ1.2) What approaches have been suggested in order to integrate usability and agile methods?	Not informed
	Answers to research questions Analyzing some works, the authors raised some issues, which were called key points of tension for integrating usability and agile development. Responding to the first research question (ARQ1.1) four main points were identified. The first point concerns the difference of focus given to those involved in the different approaches, where the agile approach focuses on the client and usability focuses on the end user. The second point concerns the focus of software delivery priority between the different approaches and their consequences. Agile approaches focus on delivering functional software, but this does not mean that it will be usable software. With this, the role of designers, whether of interface or user experience, ends up being out of context. The third point concerns the focus given by each approach in the design and modeling phase of systems. Approaches related to usability issues typically adopt up-front design while agile approaches avoid this, and initiate design whenever required by the customer. The fourth point identified concerns the issue related to perform software tests. Agile approaches focus on executing functional testing, unit testing, and acceptance testing. However, most of the time agile practitioners do not have time to perform usability tests, which impairs the usability of the final product. In response to the second research question (ARQ1.2) the authors also identified several approaches, proposed by other authors, to integrate usability into agile development. Among them: use of techniques such as discount usability engineering [33], integration through user-centered design [33, 35, 36, 37], proposal to include usability tests [38], use of scenarios [5, 39], combining extreme programing and usability engineering [40].		
ACM	Silva et al. [S2]	They conducted a systematic literature review of existing literature related to the integration of agile software development with user-centered design approaches. The goal of this study was to identify existing evidence, including practices and artifacts, regarding the integration of UCD and Agile to support a proposal of a methodology.	(ARQ2.1) How are usability issues addressed in agile projects? (ARQ2.2) What are common practices to address usability issues in agile methods?	58 primary studies
	Answers to research questions Through the analysis of the 58 papers selected in this systematic review the authors classified them regarding the content, considering their approach and research method. Regarding to approach they were classified as: *specialist* if the team used specialists for UCD work, *generalist* if all team members fulfill both roles and *generalist/specialist* if some development team members fulfill both approaches, but not all. Regarding to research method they were classified as: *experimental, empirical, experience report* and *theoretical*. The studies were also classified according to the techniques used by the teams. This study identified several ways to address usability in agile projects (ARQ2.1). Among them, the use of some artifacts and practices such as little design up-front, close collaboration, low fidelity prototypes, user stories, inspection methods, one sprint ahead, big picture, interactions models, guidelines and essential use case were identified. To answer the second research question (ARQ 2.2) the authors classified the related works and quantified the results. The main practices identified were: little design up-front, prototyping, user stories, user testing, inspection evaluation and one sprint ahead. This review also showed a clear need for more experimental studies regarding UCD and agile methods, because the results showed that 21 papers were theoretical, 20 were empirical, 19 were experience reports and only one paper was classified as experimental.		
ACM	Barksdale and McCrickard [S3]	This study is a literature review and aims to address the interaction-related problems in agile usability teams. The authors explored how social capital and social network governance may contribute to effective management of usability knowledge in agile usability software teams. They also intended to offer some practical guidance on designing cohesive agile usability teams.	Not defined	65 primary studies
	Answers to research questions In this study no research questions were defined but the authors presented some contributions as specific goals to be achieve. The first one is a background, analysis and discussion of the various agile usability integration strategies. The second one is related to a model for investigating the social interaction in agile usability teams. The third is a research, practice, and policy agenda for future work toward improving the social interaction in multidisciplinary agile usability software teams. The authors identified five categories of agile usability integration that were researched or used, represented by: practices, processes, technology, people and social. This study explored the social category discussing the potential impact of not considering it in agile and usability integrations and criticized the role of social interaction in the other four categories. This was possible because the authors created a framework for evaluating the social interaction and usability knowledge management in the integration of agile and usability teams. Authors also suggested an additional study of the social interaction in agile usability software teams, as an agenda for future work.		
SpringerLink	Wale-Kolade, Nielsen and Päivärinta [S4]	This study is a systematic literature review that aimed to show how the previous studies related to integrate usability work into agile software development provides grounds, warrants, backing, rebuttal, and qualification by analyzing their claims.	(ARQ4.1) What are the recommendations on how usability work should be executed within agile contexts? (ARQ4.2) Are there situational factors that influence these, and what is the nature of such influences?	49 primary studies
	Answers to research questions To answer the first research question (ARQ4.1) the researchers classified the selected papers through the lens of the Toulmin model [41] that provides a concise framework to assess the argumentative structure of the identified recommendations. A content analysis was applied to identify themes related to the Toulmin model which led to the identification of common recommendations, and their grounds, rebuttals and qualifiers. Seven claims were identified: (1) Conduct some upfront design activities prior to project start; (2) Design low-fi prototypes as the basis for developing the system; (3) Perform testing between iterations; (4) Designers and developers each work in parallel; (5) Usability designers should be a part of the development project; (6) Usability designers should be fully integrated into the development team; (7) End users or their proxies should be involved in the project life cycle,		

(*continued on next page*)

Table 4 (*continued*)

Database	Authors	Goal	Article research questions	Number of studies
	and answer the first research question. To answer the second research question (ARQ4.2) another step was executed, and all identified claims and their qualifications were recategorized based on Clarke and O'Connor [42] framework of situational factor (business, personnel, requirements, operation, organization, application, management and technology) which can influence the software development process. Almost all qualifications were addressed to a situational factor except "technology." The existence of this gap needs to be considered by agile usability research to understand how the technology employed could affect the identified claim.			
ACM	Salvador, Nakasone and Pow-Sang [S5]	The goal of this study was to present the results of a systematic review involving the use of usability techniques in software development where agile methodologies were used.	(ARQ5.1) Which usability methods have been applied in agile software? (ARQ5.2) In which phases or artifacts of agile software development have usability methods been applied? (ARQ5.3) Which kinds of evaluations have been performed when using usability methods in agile software development? (ARQ5.4) Which empirical studies regarding usability methods have been applied in agile software development?	32 primary studies
	Answers to research questions In order to answer the four research questions, the 32 selected papers were classified according to 4 factors: usability technique (*inspection, testing, inquiry, complementary*), phases in which the artifacts were applied (*requirements, design, implementation*), types of evaluations that were performed (*manual or automated*), type of empirical study that was performed (*surveys, case studies, experiments, no study*). According to the results obtained, a complementary technique such as fast prototyping, was the most frequently usability method used, corresponding to almost 69% of the studies. The application of the usability methods was performed during the implementation phase according to 50% of the selected studies, which is the riskiest in terms of software modifications costs. Only 10% of the studies used some kind of automated tool, which may definitely help the reduction of additional time when implementing any usability method during the software development. Around 6% of the studies performed surveys and the majority of the studies performed case studies. This study highlighted that very few controlled experiments have been performed and there is a need for more studies to understand how to better integrate usability into agile software development.			
ACM	Salah, Paige and Cairns [S6]	This study is a systematic literature review that aims to identify challenging factors that restrict the integration of Agile and User Centered Design. During the study the authors explored some proposed practices to deal with the identified challenging factors.	(ARQ6.1) What are the challenges that could develop during AUCDI (Agile and User Centered Design Integration) adoption process? (ARQ6.2) What are the potential success factors for AUCDI? (ARQ6.3) What are the potential practices for AUCDI?	71 primary studies
	Answers to research questions According to this study seven challenges were identified related to agile and user centered design integration: (1) lack of time for upfront activities, (2) difficulty of modularization/chunking, (3) difficulty of prioritizing UCD activities, (4) optimizing the work dynamics between developers and UCD practitioners, (5) performing usability testing, (6) UCD practitioner workload, (7) lack of documentation. For each of the challenges identified the authors proposed some practices to deal with the identified challenging factors and to transform them in potential success factors. For example, for the first challenge the authors proposed to include up-front design, that is a separate pre-development period used in agile projects for eliciting requirements referred to as "Iteration 0." The second was addressed via having well defined design goals, chunking design into features and postponing depth based UX activities to occur later in the development life cycle. The third was handled via assigning this responsibility to the designer or UCD practitioner and have a separate UX backlog. The fourth challenge was handled by sharing an understanding of users and design vision, and also synchronizing efforts of UCD practitioners and developers. The fifth was handled via preparation for user research utilizing discount usability engineering techniques including: heuristic evaluation, Rapid Iterative Testing and Evaluation (RITE) and using low fidelity prototypes. The sixth was addressed via conducting mentoring process to developers and distributing UCD practitioner workload on two team roles; a UCD researcher and an UCD prototype. The last challenge was addressed through documenting via wikis, use cases, scenarios, personas, wire frames, sketches, design patterns, and tool support.			
ACM	Jurca, Hellmann and Maurer [S7]	In this study the authors performed a systematic mapping study to identify relevant research studies related to integration of agile software engineering, user-centered design and user experience. The goal of this study was to understand what the field of Agile-UX looks likes at the present.	(ARQ7.1) Is the rate of publication increasing over time? (ARQ7.2) What venues are most important for this field? (ARQ7.3) What types of papers are most prevalent? (ARQ7.4) Are the types of studies changing over time? (ARQ7.5) What are the recommendations of existing work?	76 primary studies
	Answers to research questions According to this study the number of publications in this field increased significantly since 2002, especially from 2007 onwards, with no year since then having fewer than 6 publications, until 2013. The Agile Conference was identified as the most important venue with 20 publications followed by CHI with 10. Regarding to the types of research conducted, the authors classified the papers according to a classification system defined by Wieringa [43] that includes the following categories: evaluation, validation, solution, experience, philosophical, opinion. The experience report was identified as the most frequent category, with 28 papers out of 76. Adding the values of the papers classified as evaluation, with 15 papers, and validation, with 7 papers, it can be stated that only 22 articles were rigorously evaluated. This result only confirms data from previous research that there are few rigorously conducted studies in Agile-UX. The authors also classified the evaluation and validation studies to verify what type of empirical research was used to investigate Agile-UX. It was clear that the researches also seemed to use an increasing amount of different type of studies within a single paper but methods such as focus group and action research were not well explored until 2013. Regarding to the recommendations the authors suggested the researches to consider the organizational context of the industry subjects they are studying. This is because Agile-UX methodologies may not be implemented without the appropriate organizational support. Some artifacts and practices were also recommended by the evaluation and evaluation studies like: concept maps, cognitive walkthrough variants, workshops, low-fi prototypes, interviews, scenarios, and meetings with users.			

(*continued on next page*)

Table 4 (*continued*)

Database	Authors	Goal	Article research questions	Number of studies
SpringerLink	Zapata [S8]	In this study the author developed a systematic literature review to answer how agile methodologies and techniques of usability have been integrated during the various stages of software development.	(ARQ8.1) What usability methods are integrated into software development methodologies? (ARQ8.2) What agile methodologies have integrated usability techniques throughout the complete software development process? (ARQ8.3) What new frameworks or methods have been proposed for the integration of agile processes and usability engineering?	37 primary studies
	Answers to research questions According to the author some methods like user-centered design, usability design, user experience design, usability evaluation and informal cognitive walkthrough were used in the primary studies selected to integrate usability to agile software development. The results showed that 59.5% of the studies applied user-centered design. The author also reported that agile methodologies as DSDM, XP, Scrum and Lean were also used by the primary studies and most of them (45.95%) proposed a new framework to integrate usability to agile software development. The author also identified that some research methods like case studies, literature and systematic reviews and exploratory studies were used by these studies. Through the analysis of the exploratory studies the author could affirm that the integration of agile methodologies and usability is convenient and totally possible.		
ACM	Brhel et al. [S9]	In this study the authors developed a systematic review focused in capturing the current state of the art in user-centered agile software development (UCASD) approaches. The goal of this study was to investigate these approaches and to derive generic principles from them.	(ARQ9.1) Which principles constitute a user-centered agile software development approach?	83 primary studies
	Answers to research questions According to the author five principles could constitute a user-centered agile software development approach: (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*. This study contributed to the software development body of knowledge by providing a broad review of the literature in the area, making a classification of the primary studies to provide a form coding system and identifying generic principles of UCASD.			
SpringerLink	Silva et al. [S10]	In this study the authors presented a systematic mapping of agile user-centered design publications at the major agile and human-computer interaction (HCI) conferences. The goal of this study was to present a summary of the Agile UCD field and to find out the topics this field encompasses.	(ARQ10.1) What is agile UCD? (ARQ10.2) What types of HCI techniques have been used to integrate agile and UCD? (ARQ10.3) What types of studies on agile UCD have been published? (ARQ10.4) What types of research methods have been used in agile UCD studies? (ARQ10.5) What benefits do these publications offer? (ARQ10.6) Who are the major authors in this field? (ARQ10.7) Is this field driven by academics, practitioners or collaborations?	46 primary studies
	Answers to research questions In this study the authors restricted their initial search for papers related to the major agile conferences (Agile, XP, XP/Agile Universe) and HCI conferences (CHI – Conference on Human Factors in Computing Systems and UIST – User Interface Software and Technology Symposium). Although we expected a more objective and descriptive answer to the first question of research, the authors classified the primary studies according to the iteration design stages to give us an idea of which phase the studies were most focused on. The stages used were: establish requirements (research), (re)design, build an interactive version (prototype) and evaluate. The authors found emphasis on research phase, however, the numbers among the other phases are relatively balanced. The authors also identified in the primary studies that the main usability technique that have been used to integrate agile and UCD is lightweight prototype. Some other techniques like evolutionary prototyping, upfront design, continuous design and personas respectively come afterwards. The main type of study on agile UCD that have been published were experience reports and the main types of research were case study and grounded theory. The main reported benefit mapped in the primary studies was improved communication followed by improved usability, improved visibility and improved analysis. The authors also enumerated the top 14 authors for the field, which may be interesting for those who are getting started in this area. Another interesting point of the analysis was the mix of academic and industry authors collaborating.			
IEEE	Dhandapani [S11]	This study intended to review some of the existing literature aiming to find out common observations and differences recorded regarding the integration of user centered design and agile approach.	(ARQ11.1) Is the integration of UCD and agile possible? (ARQ11.2) Is the team of UI designer and developer able to successfully integrate and deliver the product? (ARQ11.3) Will the team repeat the model they found?	Not informed
	Answers to Research Questions Regarding to the first article research question (ARQ11.1) the author identified that the integration of UCD and agile processes was possible, with the right planning. The primary studies indicated that their integration was successful, despite some difficulties, and the final products were, in general, delivered with better quality. To answer the second research question (AQR11.2), the author identified that almost all papers have reported that the team as a whole had a better cohesiveness about the product, at the end of an integration release, and better confidence on the finished product. Regarding to the re-use of the model under study (ARQ11.3), the team also had a cohesive response on the positive factors of the integration result that surpassed the initial challenges that could have happened during the first stages of the integration and therefore would repeat the model under study.			

(*continued on next page*)

Table 4 (*continued*)

Database	Authors	Goal	Article research questions	Number of studies
IEEE	Magües et al. [S12]	In this study the authors conducted a systematic mapping study to investigate the integration of the agile software development and user-centered design according to four criteria: processes, practices, team and technology.	(ARQ12.1) What is the current state of the integration of agile processes and usability?	161 primary studies
Answers to Research Questions In this study authors classified the primary studies according to four criteria: process, practice, team and technology integration. To summarize the results a Venn diagram was used to make clear the interconnections among the areas using the theory of sets. The analysis revealed that practice and process integration was strongly represented (with 76 publications) and the least representative category was supporting technology integration (with only 7 publications). The study also revealed that the agile community is very much interested in the adoption of usability techniques, however, none of the studies proposed generally applicable guidelines for the agile community to follow.				
IEEE	Magües et al. [S13]	In this study authors conducted a systematic mapping study to investigate the integration of the agile software development and user-centered design throughout usability techniques.	(ARQ13.1) What is the current state of the integration of agile processes and usability techniques?	31 primary studies
Answers to research questions In this study authors classified the 31 primary studies selected according three stages of software engineering development process: requirements engineering, design and evaluation to determine all techniques that were reported during these stages by the paper authors. According to the analysis Persona is the most commonly adopted technique in requirements engineering activities followed by Contextual Inquiry and Scenarios. The Screen Snapshot is a prominent technique in design activities and for evaluation stage the usability test is the most commonly adopted. The authors created a table to describe all classifications made and also indicated if the technique was used "as-is" or "with modifications" or adaptations.				
ACM	Bertholdo, Kon and Gerosa [S14]	In this study a literature review was conducted to identify patterns of use of agile usability practices focusing on the user centered design final stages.	(ARQ14.1) What are the agile usability practices related to the final stages of UCD used?	Not informed
Answers to research questions In this study authors analyzed the primary selected studies and focused on identifying usability practices related to two specific steps of the UCD: (1) Create Design Solutions, (2) Evaluate Design, to become patterns. If the practice identified was used by at least three different cases it became candidate to be a pattern. Their presentation structure was described through: name, context, problem, solution and examples. As a result of this study eight patterns were described, four related to Creative Design Solutions: (1) Low Fidelity Prototyping, (2) High Fidelity Prototyping, (3) Design Studio, (4) Collaborative and Participative Design, and four related to Evaluate Design: (5) Tests with Users, (6) Evaluation by Inspection, (7) RITE Method, (8) Acceptance Tests.				

A) Classification according to the scores

In general, the results, presented in Table 7, tell us that the studies related to systematic literature reviews were better elaborated and were more rigorous in relation to quality criteria compared to the others. If we analyze the results in isolation, considering that works with high quality have scores between 3.5 and 5, medium quality between 2 and 3 and low quality between 0 and 1.5 we can see that the number of studies with high quality is still very low. Only four out of fourteen studies had the score equal or greater than 3. The majority of the studies were concentrated on medium or low-quality classification with six studies in the range 2.0 to 3.0, and four had a score below 2.0. Only one study presented by Brhel et al. (2015) was scored with 5 points.

B) Classification according to quality criteria

Only one of the secondary studies presented quality criteria's and assessed the quality of the primary studies. According to Petersen et al. (2008), this is expected in systematic mapping studies, but it can bring problems related to the rigor and relevance of the primary studies in systematic literature reviews. Comparing the final scores of the systematic literature reviews we noticed that only one study has graded the maximum score. This study presented all the 5 criteria analyzed by the quality assessment.

The second criterion less scored in the systematic literature reviews was the one related to synthesis method. Two out of five studies did not mention or make references to a synthesis method in the systematic literature reviews studies. This deficiency also appeared in the mapping studies and non-systematic reviews. This points us to some weaknesses that need to be worked out in future similar works.

The adequacy of search in the systematic mapping studies was also graded with low scores. This demonstrates that perhaps some important studies in the area may not have been considered in the research leading to a bias in the research results.

What really caught our attention were the low scores found in non-systematic literature review studies. Although these studies do not follow any defined protocol, it is expected to find at least one criterion of selection of the studies, the search database or the method used to synthesize the results. None of them were found.

RQ4) What challenges are described in the published studies related to the integrations of usability and agile software development?

Sohaib and Khan [S1] focus their work on identifying the tensions between usability and agile methods. It is possible to say that these tensions may be some kind of challenge to the integrations of usability and agile methods, as following:

- The different focus on customer versus end-user. In agile software development the customer represents a role with many responsibilities and its participation in the process of participatory design and collaboration are not equal as the real users. As they usually do not represent typical users, they do not ensure that the application will be usable.
- The different focus on software delivery. While the agile software development aims to deliver functional software as soon as possible the UCD has the commitment to deliver usable software.
- The different views of the application design. While UDC processes rely on up-front user design, agile practitioners usually try to avoid an extensive up-front design phase.
- The focus of each type of test that is applied. Different types of tests are emphasized in agile methods, like unit testing and acceptance testing, but usually tests related to usability are neglected. Probably issues related to the time (short sprints) in release cycles force them to emphasize the unit test and acceptance test with no directly support to usability test.

Table 5
Main ways to integrate usability to agile software development.

Authors	Ways of integration	Processes associated	How to achieve the integration
Sohaib and Khan [S1]	A combined approach	Usability engineering and agile methods	1) Iterative development 2) Assemble a multidisciplinary team to ensure complete expertise 3) Collaboration among customers, users, product managers, business analysts and developers 4) A combination of test (unit testing, user acceptance testing and usability testing) 5) An integration of user stories with scenario-based design
Silva et al. [S2]	A new framework that included activities and artifacts	Agile software development and user-centered design	1) Works with parallel iterations, one for developer and another for designers 2) Include the concept of "n Sprints", and usability evaluations 3) Include evaluations using low-fi prototypes and constant feedback
Barksdale and McCrickard [S3]	Different approaches (practices, process, technology, people and social)		1) The adoption of practices by one or both areas 2) The combination of agile and usability processes 3) The communication between technology used in each domain 4) The incorporation of team members with the necessary skill 5) The integration with greater focus on their interaction
Wale-Kolade, Nielsen and Päivärinta [S4]	Claims (about how to integrate usability work into agile development)		1) Conduct some upfront design activities prior to project start 2) Design low-fi prototypes as the basis for developing the system 3) Perform testing between iterations 4) Designers and developers each work in parallel 5) Usability designers should be part of the development project 6) Usability designers should be fully integrated into the development team 7) End users or their proxies should be involved in the project life cycle
Salvador, Nakasone and Pow-Sang [S5]	Usability techniques		1) Prototyping 2) Formal tests with real users thinking aloud 3) Individual inspections (through questionnaires and/or surveys) 4) Heuristics evaluation 5) Cognitive walkthrough
Salah, Paige and Cairns [S6]	Practices (to deal with the challenges factors that restrict agile and user-centered design integration)	Agile software development and user-centered design	1) Use of upfront design by development and quality assurance teams to work on back end features or features with low design cost and high development cost 2) Postpone depth based UX activities to occur latter in the development life cycle, develop functional features and its related UX design activities in the same iteration 3) Assign this responsibility to the designer or UCD practioner's, including a separate UX product backlog 4) Share an understanding of users, share an understanding of design vision and synchronize efforts of UCD practitioners and developers 5) Use preparation for user research utilizing discount usability engineering techniques such as: heuristics evaluation, low fidelity prototypes, RITE and remote usability tests 6) Use acceptance tests and demonstration sessions or introduce a mandatory user interface review as a gate-keeping tool 7) Include planning in advance for user inclusion and conduct tests with an existing user pool to participate as developer or designer partners 8) Use UX practitioners to act as agile customers to validate the design passed to developers to implement 9) Conduct mentoring processes to developers preparing them to perform the role of UCD practitioners, decrease UCD practitioner's workload distributed in two teams roles (UCD researcher and UCD prototypes) 10) Document the projects via wikis, scenarios, personas, sketches, wire frames, webpages, use cases, prototypes, information radiators, design patterns and tool support

(*continued on next page*)

Table 5 (*continued*)

Authors	Ways of integration	Processes associated	How to achieve the integration
Jurca, Hellmann and Maurer [S7]	Organizational recommendations	Agile software development and user-centered design	1) Ensure that user experience (UX) designers have sufficient time for the projects so they can focus on accomplishing all assigned tasks 2) Assign UX designers few projects to improve the UX vision of the product 3) Have a share workspace between UX designers and developers, to encourage the team to see the UCD as an equal part of the project 4) Give organizational support to the usability specialist because the most essential aspect of the integration is the contextual value that the team is embedded in, rather than practices themselves, to ensure the success of integration
Zapata [S8]	Process and frameworks		1) In its analysis the most frequent process to integrate usability to agile software development proposed in the primary studies was user centered-design representing 59.5% of them, followed by usability evaluation with 18.9% 2) It was also noticed that a great number of the primary studies, 45.95%, proposed new frameworks for integrating usability and agile development
Brhel et al. [S9]	Five principles that were associated with specific practices and process	Agile software development and user-centered design	1) The first principle is related to separate product discover and product creation 2) The second is related to iterative and incremental design and development 3) The third is related to include parallel interwoven creation tracks 4) The fourth is related to the continuous stakeholder involvement 5) The fifth is related to artifact-mediated communication
Silva et al. [S10]	Some types of HCI techniques	Agile software development and user-centered design	1) The main one is usability testing on lightweight prototypes (with 13 occurrences) 2) Continuous research (with 11 occurrences) 3) Expert evaluation was identified (with 6 occurrences) 4) Evolutionary prototyping, upfront design, continuous design, personas, were all identified in the primary studies (with 6 occurrences) 5) Automated usability evaluations, user stories and RITE -Rapid Iterative Testing and Evaluation (with 5 occurrences)
Dhandapani [S11]	Common patterns	Agile software development and user-centered design	1) Keep the user centered designer (UCD) one sprint ahead of the actual team developer to ensure the deliverable to be useful and usable 2) Include iteration and feedback to allow the incorporation of the UCD into the system 3) Evaluate the output of UI designers from a sprint through usability tests before to pass it to the developers 4) Test the deliverable from developers for acceptance with customers
Magües et al. [S12]	Process integration, practice integration, team integration and technology integration		1) Their analysis reveals that practice and process integration are strongly represented in the analysis. Process integration was represented by 76 out of 161 studies, which represent 47.83% of the total 2) Practice integration is the second largest group with 31 publications, representing 19.25% of all primary studies analyzed 3) The team integration category is the third largest group composed of 27 primary studies (16.77%) 4) The least representative category is supporting technology integration with only seven primary studies (4.34%) 5) 11.18% of the primary studies are positioned around the category intersections
Magües et al. [S13]	Some HCI techniques	Agile software development and user-centered design	1) They found that personas were the most adopted technique in requirement engineering activities 2) The second most popular technique is contextual inquiry followed by scenarios and prototyping 3) Screen snapshots are also a prominent technique in design activities 4) Finally usability testing is said as the most commonly adopted technique within evaluation activities
Bertholdo, Kon and Gerosa [S14]	Agile usability practices based on literature in form of patterns focusing on the UCD final stages	Agile software development and user-centered design	1) For the first step (Create Design Solutions) four patterns were proposed: low fidelity prototyping, high fidelity prototyping, design studio, collaboration and participative design 2) For the second step (Evaluate Design) four patterns were also proposed: test with users, evaluate by inspection, RITE method, acceptance test

Table 6
Main categories identified in the secondary studies for the integration of usability to agile software development.

Main categories of integration	Main forms of integration	Authors
Processes	A combined approach (usability engineering and agile methods)	Sohaib and Khan [S1]
	User-centered design	Zapata [S8], Silva et al. [S10], Salah, Paige and Cairns [S6], Jurca, Hellmann and Maurer [S7], Dhandapani [S11], Magües et al. [S13], Bertholdo, Kon and Gerosa [S14], Brhel et al. [S9]
Techniques	Some usability techniques	Salvador, Nakasone and Pow-Sang [S5]
	Some types of HCI techniques	Silva et al. [S10]
	Some HCI techniques	Magües et al. [S13]
Practices	Some practices (to deal with the challenges factors that restrict agile and user centered design integration)	Salah, Paige and Cairns [S6]
	Agile usability practices based on literature in form of patterns focusing on the UCD final stages	Bertholdo, Kon and Gerosa [S14]
	Common patterns	Dhandapani [S11]
	Claims (about how to integrate usability work into agile development)	Wale-Kolade, Nielsen and Päivärinta [S4]
Recommendations	Organizational recommendations	Jurca, Hellmann and Maurer [S7]
Principles	Five principles that were associated with specific practices and process	Brhel et al. [S9]
Different approaches	Process integration, practice integration, team integration and technology integration	Magües et al. [S12]
	Different approaches (practices, process, technology, people and social)	Barksdale and McCrickard [S3]
	A new framework that included activities and artifacts	Silva et al. [S2]
	New frameworks for integrating usability and agile development Frameworks	Zapata [S8]

Salah, Paige and Cairns [S6] identified through a systematic literature review various challenges factors that restrict agile and user-centered design integration but also explored and proposed practices to deal with it, such as:

- Lack of time to develop upfront activities.
- Difficulties of design modularization, called as design chunks. To break design into cycle sized pieces is not natural as it is in agile development, which makes this process critical and challenging.
- Difficulty of prioritizing UCD activities due to developers' focus on accomplishing functionality features.
- Optimizing the work dynamics between developers and UCD practitioners. As the UI designers have become part of the development team, they need to be more present and actively participate in the project discussions.
- Perform usability tests that include various sub-challenges such as: scheduling usability testing, accessing users for usability testing, methods of usability testing, and shorter time to iterate design.
- UCD practitioners' workload. Usually UCD practitioners are used as shared resources and the lack of dedicated resources for each project can impose an extra burden.
- Lack of documentation. Although agile methodologies strive to work with minimal documentation, it is critical for a good integration between agile methods and usability. The lack of documentation can lead to a lack of understanding of requirements by the team, as well as deliveries and other associated tasks, such as tests, delivery dates, and the actual progress of the project.

Jurca, Hellmann and Maurer [S7] discussed some organizational problems that can hinder the agile software development and UX integration:

- The power struggle between UX designers and developers. Several organizations described that UX designers are not considered as full-time member of the agile team [44], and in other organizations there are not enough UX designers involved and working in agile projects. As a result, UX designers are always working in more than one project at a time as shared resources and become overloaded.
- Another problem occurs when developers, UX designers and stakeholders lack of a UX vision of the whole product making the project slowing down.

Bertholdo, Kon and Gerosa [S14] also presented some problems related to the integration of usability to agile software development:

- The usage of low fidelity prototyping for the key features of entire systems prior to development. This practice was largely used in traditional development, but it has become incompatible with agile environments, as its development cycle comprises a series of small incremental releases.
- Get feedback from customers and users to evaluate the interface and the interaction flow prior to development. This practice is also incompatible with agile environments, as the feedback process can be developed during the development process.
- The short time of each iteration to exploit several design options.
- Participation of both roles (developers and designers) in the process of creating design solutions. When only one team is involved with the creation of alternative design versions, the understanding of the whole solution became deficient.
- Evaluate the systems with real users. This was an essential usability practice in traditional development, but in agile iteration, with less time to run evaluations, it is not a trivial activity.
- Evaluation by inspection. It is another important task executed in traditional development to refine the usability of the system. However, in agile iterations the inspection is done in each iteration because there is no much time to execute and correct the problem.
- Lack of time for usability test and corrections problems.

Table 7
Quality assessment of selected studies.

Type of research	Authors	Quality criteria Inclusion/exclusion criteria	Adequacy of search	Synthesis method	Quality criteria	Information provided about primary studies	Final Score
SLR	Silva et al., [S2]	1	1	0.5	0	1	3.5
SLR	Wale-Kolade, Nielsen and Päivärinta, [S4]	0	0	1	0	1	2
SLR	Salvador, Nakasone and Pow-Sang, [S5]	1	1	0	0	0	2
SLR	Salah, Paige and Cairns, [S6]	1	1	1	0	1	4
SLR	Zapata, [S8]	1	0.5	0	0	0.5	2
SLR	Brhel et al., [S9]	1	1	1	1	1	5
SMS	Jurca, Hellmann and Maurer, [S7]	0	1	1	0	1	3
SMS	Silva et al., [S10]	1	0.5	0	0	0.5	2
SMS	Magües, Castro and Acuña, [S12]	1	0.5	0	0	1	2.5
SMS	Magües, Castro and Acuña, [S13]	1	0.5	1	0	1	3.5
NSLR	Sohaib and Khan, [S1]	0	0	0	0	0.5	0.5
NSLR	Barksdale and McCrickard, [S3]	0	0	0.5	0	0	0.5
NSLR	Dhandapani, [S11]	0	0	0	0	0.5	0.5
NSLR	Bertholdo, Kon and Gerosa, [S14]	1	0	0	0	0	1

In Table 8 we grouped the challenges mapped in the selected studies in order to identify the main challenges reported and their occurrences. After grouping the main challenges, it was possible to identify seven main categories: issues related to tests, time, work balance, modularization, feedback, prioritization, and documentation.

5. Discussion

This tertiary study was conducted by following the guidelines proposed by Kitchenham and Charters [26] to conduct systematic literature reviews. Of the initial 3065 papers selected in well-known electronic research databases, 14 studies were selected following a rigorous process, from the selection of the studies to the discussions performed to solve disagreements found during the selections process that were performed in pairs. Although we did not delete any research, all 14 selected studies underwent to a quality assessment, which were presented in Table 7. The execution of this whole process besides providing a systematic way that can be reproduced by other researchers, also gave more confidence on the resulting analysis.

As our goal was to present a critical analysis of the published secondary studies, we noticed some anomalies and potential conflicts in the reported number of primary studies in the systematic literature reviews on the same topic by different authors in different years. We noticed it because the number of studies selected on the systematic reviews, even presented in a chronological order, does not increased as presented in Fig. 4.

The first published secondary study as a systematic review was in 2011 [S2] and it analyzed 58 primary studies. The second systematic review was published in 2013 [S4] and analyzed 49 studies. The third and fourth ones were also published in 2014, [S5] analyzed 32 studies and [S6] analyzed 71 studies. In 2015 [S8] analyzed 37 primary studies and [S9] analyzed 83 primary studies. It is interesting because when we analyzed the adequacy of the research, for systematic literature reviews, we noticed that the majority of the studies selected three or more database search, including IEEE Explore Digital Library, ACM Digital Library and SpringerLink, and usually with a research string very close to each other. Besides that, usually in systematic reviews the authors created a session to describe previous works, including secondary studies, and also justified the reasons for conducting a new systematic review. However, in our analysis, these issues were not found. For example, Salvador, Nakasone and Pow-Sang (2014) analyzed 32 primary studies and Silva et al. (2011) analyzed 58 studies and we also noticed that Silva et al. (2001) work was not even referenced by Salvador, Nakasone and Pow-Sang (2014).

Examining our third research question it was also possible to point out some weaknesses of the selected secondary studies. The less graded criterion in the analysis, not only for the analysis of the systematic literature reviews, was related to quality. This means that there was no pre-established quality criterion to filter the studies during the selection process. So, this may bring a bias to their results.

It was also possible to observe that some of the selected studies [S7], [S4] used a prior systematic literature review published by Silva et al. (2011) as a starting point of their research. As in his work no quality criterion was established, a bias could be propagated. Thus, it is important to have previous published works as references but not used as cutting point because using the whole set, new results may emerge. Another source of bias can be the number of selected research databases. It was possible to verify an absence of care in the systematic mapping works as well as the non-systematic reviews in providing different (more than 3) sources of research in the selection of works.

Analyzing all the selected papers we have seen that six years have passed from the publication of the first study [S1] to the publication of the last one [S14]. In those 6 years we could verify that early studies focused on the difficulties of integrating usability into agile development, whether through usability engineering [S1] or user-centered design [S2]. It was also evidenced in early studies that usability and

Table 8
Categories of challenges identified in the secondary studies for the integration of usability to agile software development.

Category of challenges	Challenge's description	Authors
Test	More emphasis on functional tests than on usability tests.	Sohaib and Khan [S1]
	To perform usability tests: scheduling usability testing, accessing users for usability testing, methods of usability testing, and shorter time to iterate design.	Salah, Paige and Cairns [S6]
	Lack of time for usability test.	Bertholdo, Kon and Gerosa [S14]
Time	Lack of time for upfront activities.	Salah, Paige and Cairns [S6]
	Short time of each iteration to exploit several design options.	Bertholdo, Kon and Gerosa [S14]
Work balance	Optimizing the work dynamics between developers and UCD practioner's.	Salah, Paige and Cairns [S6]
	Power struggle between UX designers and developers (UX overloaded).	Jurca, Hellmann and Maurer [S7]
	Participation of both roles (developers and designers) in the process of creating design solutions.	Bertholdo, Kon and Gerosa [S14]
Modularization	Difficulties of design modularization.	Salah, Paige and Cairns [S6]
	Usage of low fidelity prototyping for the key features of entire systems prior to development.	Bertholdo, Kon and Gerosa [S14]
	Different views of the application design (up-front X avoid an extensive up-front design phase).	Sohaib and Khan [S1]
Feedback	To get feedback from customers and users to evaluate the interface and the interaction flow prior to development.	Bertholdo, Kon and Gerosa [S14]
	To evaluate the systems with real users.	Bertholdo, Kon and Gerosa [S14]
	Different focus on customer versus end-user.	Sohaib and Khan [S1]
Prioritization	Difficulty of prioritizing UCD activities.	Salah, Paige and Cairns [S6]
	Different focus on software delivery (functional x usable).	Sohaib and Khan [S1]
	Lack of a UX vision of the whole product.	Jurca, Hellmann and Maurer [S7]
Documentation	Lack of documentation.	Salah, Paige and Cairns [S6]

agile development were being worked in separated ways, segregating roles and responsibilities into specific tracks with parallel work. Over time some frameworks have been proposed [S2], [34,45] but there is still room for new proposals and guidelines to explore this integration. We did not find approaches that explored: practices to really anticipate end user's problems before starting the sprints, and not just using sprint zero to solve them; a mutual understanding at the beginning of the project about the actual user journey and what the product should be, including technical, business and user experiences perspectives; and guidance for identifying critical user's iteration points conduction usability and user experience tests. It was also possible to perceive that not only usability has influenced the selected researches but also the concern with the final user experience [S7]. The integration of both issues (usability and user experience) with agile methods, through user-centric approaches, was called Agile-UX. The idea is to distribute the activities related to the combination of these issues throughout the entire development process and the trend is no longer manage and control these activities in separate ways. Although we are walking towards this junction, there are several aspects that still cause concerns, ranging from issues linked to organizational cultures to support and maintain such activities, through difficulties with the team itself, allocation of resources, division of activities, appropriate documentation and artifacts.

6. Threats to validity

The first threat to 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 [26] and Petersen et al. [27] were followed. Using the criteria to studies selection phase, we excluded the work of Cockton et al. [46] once it is a kind of proceedings of the workshop that was held in 2014 (NordiCHI) and not a secondary study itself.

This systematic study was developed using four databases (SpringerLink, ScienceDirect, ACM and IEEExplore) 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 run several tests before deciding to use this search string because we faced some problems using IEEExplore database, which in that period was limited to 15 terms to run the search string. As we intended to run the same string in all databases, we decided to keep it within this limited size. This lead us to make several string tests before the final decision, to validate that the most relevant works already found were being returned by the final search string.

The second point is related to the quality assessment process. To decrease the risk of bias in the quality assessment process we decided to make this analysis in pair. Therefore, two researchers performed the quality assessment independently and the results were discussed to

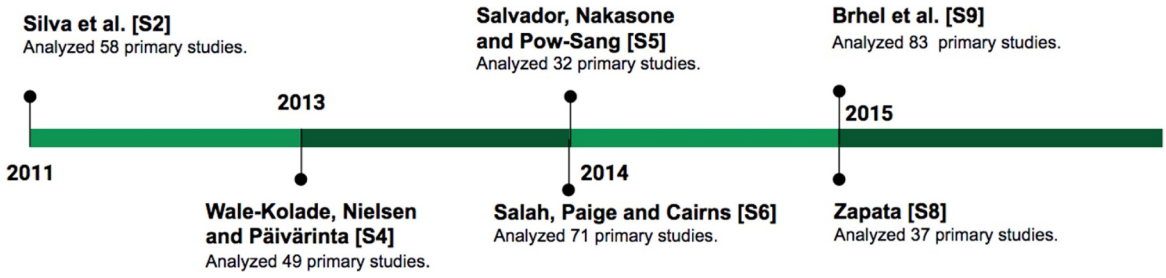

Fig. 4. Number of primary studies analyzed by systematic reviews in a chronological order.

reach consensus.

The third point is related to the type of studies included in this analysis. As we decided to include systematic mapping studies and non-systematic literature reviews to complement the results of this study and knowing about the differences in scope, process and the depth of the studies, the results were presented by grouping the different types of studies as demonstrated in Table 5.

Even including different types of studies, the final number of studies focusing on explore the integration of usability and agile software development is still low, comparing to others tertiary studies [32,47,48]. In our analysis 14 secondary studies were found that may be used as a starting point for future researches.

7. Conclusion and future work

This section will present the final conclusions of the study, taking into consideration the research questions that based this study.

1) **What research questions were investigated in the secondary studies?**

Table 4 describes in detail all research questions used on selected secondary studies. In general we have some studies investigating the key point of tension or challenging factors to integrate usability and agile methods [S1], [S3], [S6] in order to provide effective usable software systems, other studies investigating the integration of agile software development and user-centered design [S2], [S7], [S9], [S10], [S11], [S12], [S13], [S14] and also some studies investigating how to integrate usability work, or usability techniques into agile software development [S4], [S5], [S8].

We also noticed that the integration of usability with agile software development is strongly aligned with user-centered design. Most studies are interested in investigating this relation and the feasibility of this integration, but there is no consensus established about how to integrate them and many studies are exploring this gap to present new approaches.

2) **What are the main ways to integrate usability and agile software development according to the secondary studies?**

During the analysis we found 14 studies, including 6 systematic reviews, 4 systematic mapping studies and 4 non-systematic reviews. As demonstrated in Table 1 the interest in researching the integration of usability and agile software development is increasing in the last years. It was possible to observe that many studies are bringing new proposals to integrate these areas. Three studies [S3], [S9], [S12] out of fourteen, explored four classification criteria of the primary studies that were based on: process integration, practices integration, team integration and technology integration. The relation between the four classification criteria was very well represented in [S12] by a Venn diagram to summarize the results of the classification. This classification gives us a notion of the size of integration possibilities and the areas that can be explored with different proposals.

After analyzing the selected studies, it was decided to group the ways of integrating usability into agile development into generic main categories. From this analysis six categories were identified: processes, techniques, practices, recommendations, principles and different approaches that were described in Table 6.

3) **What are the indicators of the quality of the secondary studies?**

After the analysis of the selected studies we could conclude that, in general, the quality criteria were frequently neglected by the studies. According to Cruzes and Dybå [32] “the process of quality appraisal is important because the quality of studies or other evidence may affect both the results of the individual studies and the conclusions derived from the synthesis”. It is important to point out this fact to avoid it in future studies. Another criterion that is frequently neglected is related to the synthesis method. It is a way of drawing and reaching conclusions from a collection of studies and that can lead to deeper conclusions.

4) **What challenges are described in the published studies related to the integration of usability and agile software development?**

Regarding the challenges for the integration of usability to agile software development, we could identify seven main categories that were described in Table 8: issues related to tests, time, work balance, modularization, feedback, prioritization, and documentation. Although we have identified the main challenges reported and their occurrences, the integration does not become unfeasible. As we already presented, through this analysis we could observe that in early studies usability and agile development were being worked in separated ways, segregating roles and responsibilities into specific tracks with parallel work. Over time some frameworks have been proposed but the trend is no longer to manage and control these activities in separate ways. So new challenges are becoming to appear, ranging from issues linked to organizational cultures to support and maintain such activities, through difficulties with the team itself, allocation of resources, division of activities, appropriate documentation and artifacts.

Future studies may use and include the results of this tertiary study, particularly regarding to the quality assessment and challenges identified in this research, as a basis for conducting new studies to investigate the integration of usability and agile software development. Readers must consider that systematic reviews are limited by search date, sources and search string. Therefore, this study can be replicated in a future study. We are also interested in investigating more evidences related to the challenges, limitations and tensions between usability and agile software development. As an agenda for future research we suggest developing more researches related to the integration of usability in the agile development through tools and people (teams), because as we went deeper into the analysis of the works, we could verify that this is a point still little explored. There are other research questions that remain unsolved such as the difference of the integration process in different business scenarios and different types of software. Another point that also did not appear in the selected studies was the artifacts that are or can be used for this integration. Artifacts are very important in communication, and also assist in the decision making of the projects. Too many artifacts may lead to excessive bureaucracy which is incompatible with agile principles, but in the other hand, no artifacts can lead to lack of means to promote the integration. Finally, it is necessary to invest in deeper and higher-quality primary and secondary studies so that the results can be really useful and can bring more confidence to the target readers.

Acknowledgments

This work is partially supported by CAPES (Coordination for the Improvement of Higher Education) – Brazil (process no. 88887.147219/2017-00).

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

[S1] O. Sohaib, K. Khan, Integrating usability engineering and agile software development: A literature review, in: 2010 Int. Conf. Comput. Des. Appl., 2010: pp. V2-32-V2-38. doi:10.1109/ICCDA.2010.5540916.

[S2] T.S. Da Silva, A. Martin, F. Maurer, M. Silveira, User-centered design and agile methods: A systematic review, Proc. - 2011 Agil. Conf. Agil. 2011. (2011) 77–86. doi:10.1109/AGILE.2011.24.

[S3] J.T. Barksdale, D.S. McCrickard, Software product innovation in agile usability teams: an analytical framework of social capital,

network governance, and usability knowledge management, Int. J. Agil. Extrem. Softw. Dev. (2012). doi:10.1504/IJAESD.2012.048302.

[S4] A. Wale-Kolade, P.A. Nielsen, T. Päivärinta, Usability Work in Agile Systems Development Practice: A Systematic Review, in: H. Linger, J. Fisher, A. Barnden, C. Barry, M. Lang, C. Schneider (Eds.), Build. Sustain. Inf. Syst., Springer US, Boston, MA, 2013: pp. 569–582.

[S5] C. Salvador, A. Nakasone, J.A. Pow-Sang, A systematic review of usability techniques in agile methodologies, Proc. 7th Euro Am. Conf. Telemat. Inf. Syst. - EATIS '14. (2014) 1–6. doi:10.1145/2590651.2590668.

[S6] D. Salah, R.F. Paige, P. Cairns, A systematic literature review for agile development processes and user centred design integration, Proc. 18th Int. Conf. Eval. Assess. Softw. Eng. - EASE '14. (2014) 1–10. doi:10.1145/2601248.2601276.

[S7] G. Jurca, T.D. Hellmann, F. Maurer, Integrating agile and user-centered design: A systematic mapping and review of evaluation and validation studies of agile-UX, Proc. - 2014 Agil. Conf. Agil. 2014. (2014) 24–32. doi:10.1109/AGILE.2014.17.

[S8] C. Zapata, Integration of Usability and Agile Methodologies: A Systematic Review, in: A. Marcus (Ed.), Des. User Exp. Usability Des. Discourse, Springer International Publishing, Cham, 2015: pp. 368–378.

[S9] M. Brhel, H. Meth, A. Maedche, K. Werder, Exploring principles of user-centered agile software development: A literature review, Inf. Softw. Technol. 61 (2015) 163–181. doi: https://doi.org/10.1016/j.infsof.2015.01.004.

[S10] T. da Silva, F.F. Silveira, M.S. Silveira, T. Hellmann, F. Maurer, A Systematic Mapping on Agile UCD Across the Major Agile and HCI Conferences, in: O. Gervasi, B. Murgante, S. Misra, M.L. Gavrilova, A.M.A.C. Rocha, C. Torre, D. Taniar, B.O. Apduhan (Eds.), Comput. Sci. Its Appl. – ICCSA 2015, Springer International Publishing, Cham, 2015: pp. 86–100.

[S11] S. Dhandapani, Literature Review1, in: 2015 Int. Conf. Data Softw. Eng., 2015: pp. 37–41. doi:10.1109/ICODSE.2015.7436968.

[S12] D.A. Magües, J.W. Castro, S.T. Acuña, Usability in Agile Development: A Systematic Mapping Study, Proc. XLII Conf. Latinoam. Informática. (2016). doi:10.1109/CLEI.2016.7833347.

[S13] D.A. Magües, J.W. Castro, S.T. Acuna, HCI usability techniques in agile development, in: 2016 IEEE Int. Conf. Autom., 2016: pp. 1–7. doi:10.1109/ICA-ACCA.2016.7778513.

[S14] A.P.O. Bertholdo, F. Kon, M.A. Gerosa, Agile Usability Patterns for User-Centered Design Final Stages, in: M. Kurosu (Ed.), Human-Computer Interact. Theory, Des. Dev. Pract., Springer International Publishing, Cham, 2016: pp. 433–444.

References

[1] S. Blomkvist, Towards a model for bridging agile development and user-centered design, in: A. Seffah, J. Gulliksen, M.C. Desmarais (Eds.), Human-Centered Software Engineering — Integrating Usability Software Development Lifecycle, Springer, Netherlands, Dordrecht, 2005, pp. 219–244, , https://doi.org/10.1007/1-4020-4113-6_12.

[2] L.L Constantine, Process agility and software usability: Toward lightweight usage-centered design. *Information Age*, 8, 8 (2002), 1–10.

[3] J. Ferreira, J. Noble, R. Biddle, Agile development iterations and UI design, Agile 2007 (AGILE 2007), IEEE, 2007, pp. 50–58, , https://doi.org/10.1109/AGILE.2007.8.

[4] J. Armitage, Are agile methods good for design? Interactions 11 (2004) 14–23, https://doi.org/10.1145/962342.962352.

[5] J.C. Lee, D.S. McCrickard, K.T. Stevens, Examining the foundations of agile usability with eXtreme scenario-based design, 2009 Agile Conference, 2009, pp. 3–10, , https://doi.org/10.1109/AGILE.2009.30.

[6] A. Seffah, M.C. Desmarais, E. Metzker, HCI, Usability and Software Engineering Integration: Present and Future, in: A. Seffah, J. Gulliksen, M.C. Desmarais (Eds.), Human-Centered Software Engineering — Integration Usability in the Software Development Lifecycle, Springer, Netherlands, Dordrecht, 2005, pp. 37–57, , https://doi.org/10.1007/1-4020-4113-6_3.

[7] ISO, ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: guidance on usability, 1998.

[8] ISO/IEC 9126, Software engineering – product quality – Part 1: quality model, 2001.

[9] ISO/IEC 25010, Systems and software engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - system and software quality models, 2011.

[10] D.A. Tyldesley, Employing usability engineering in the development of office products, Comput J 31 (5) (1988) 431–436 https://doi.org/10.1093/comjnl/31.5.431.

[11] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey, Human-Computer Interaction, Addison-Wesley Publishing Company, UK, 1994.

[12] J. Nielsen, The usability engineering life cycle, Computer 25 (1992) 12–22, https://doi.org/10.1109/2.121503.

[13] ISO, ISO 9241-210:2010 Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems, 2010.

[14] T. Tullis, W. Albert, Measuring the user experience, Measuring the User Experience (2013) 320 http://dx.doi.org/10.1016/B978-0-12-415781-1.00007-8.

[15] H. Sharp, Y. Rogers, J. Preece, Interaction Design: Beyond Human–Computer Interaction, 2nd ed., John Wiley & Sons Inc., West Sussex, UK, 2007.

[16] ISO, ISO 13407:1999 Human-centred design processes for interactive systems, 1999.

[17] Agile manifesto: manifesto for agile software development, Available: http://www.agilemanifesto.org/, (2001) , Accessed date: 20 April 2018.

[18] T. Dingsøyr, S. Nerur, V. Balijepally, N.B. Moe, A decade of agile methodologies: towards explaining agile software development, J Syst Softw 85 (2012) 1213–1221, https://doi.org/10.1016/j.jss.2012.02.033.

[19] A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Addison-Wesley, 2004 ISBN 0-201-69947-8.

[20] J. Stapleton, DSDM: Business Focused Development, 2nd ed., Pearson Education, 2003 ISBN 978-0321112248.

[21] S.R. Palmer, J.M. Felsing, A Practical Guide to Feature-driven Development, Prentice Hall, Upper Saddle River, NJ, 2002 ISBN 0- 13-067615-2.

[22] M. Poppendieck, T. Poppendieck, Lean Software Development – An Agile Toolkit for Software Development Managers, Addison-Wesley, Boston, 2003 ISBN 0-321-15078-3.

[23] K. Schwaber, M. Beedle, Agile Software Development With Scrum, Prentice Hall, Upper Saddle River, NJ, USA, 2001.

[24] K. Beck, C. Andres, Extreme Programming eXplained: Embrace Change, 2nd ed, Addison-Wesley, Boston, MA, USA, 2004.

[25] T. Dingsøyr, T. Dybå, N.B. Moe, Agile Software Development: Current Research and Future Directions, 1, Springer-Verlag, Berlin Heidelberg, 2010238.

[26] B.A. Kitchenham, S. Charters, Guidelines For Performing Systematic Literature Reviews in Software Engineering, Keele University, Keele, UK, 2007 EBSE Technical Report EBSE-2007-012007.

[27] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, 12th International Conference Evaluation and Assessment Software Engineering, ,, BCS Learning & Development Ltd., SwindonUK, 2008, pp. 68–77 http://dl.acm.org/citation.cfm?id=2227115.2227123.

[28] J.M. Verner, O.P. Brereton, B.A. Kitchenham, M. Turner, M. Niazi, Systematic literature reviews in global software development: a tertiary study, 16th International Conference Evaluation and Assessment Software Engineering (EASE 2012), 2012, pp. 2–11, , https://doi.org/10.1049/ic.2012.0001.

[29] B. Kitchenham, R. Pretorius, D. Budgen, O.P. Brereton, M. Turner, M. Niazi, S. Linkman, Systematic literature reviews in software engineering – a tertiary study, Inf Softw Technol 52 (2010) 792–805 https://doi.org/10.1016/j.infsof.2010.03.006.

[30] Centre for Reviews and Dissemination, What are the criteria for the inclusion of reviews on DARE? Available: http://www.york.ac.uk/inst/crd/faq4.htm, (2007) , Accessed date: 20 April 2018.

[31] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, J. Popay, Testing methodological guidance on the conduct of narrative synthesis in systematic reviews: effectiveness of interventions to promote smoke alarm ownership and function, Evaluation 15 (2009) 49–73, https://doi.org/10.1177/1356389008097871.

[32] D.S. Cruzes, T. Dybå, Research synthesis in software engineering: a tertiary study, Inf Softw Technol 53 (2011) 440–455 http://dx.doi.org/10.1016/j.infsof.2011.01.004.

[33] D. Kane, Finding a place for discount usability engineering in agile development: throwing down the gauntlet, Proceedings of the Agile Development Conference (ADC'03), IEEE Press, 2003AO-46.

[34] D. Sy, Adapting usability investigations for agile user-centered design, J Usability Stud 2 (3) (2007) 112–132.

[35] J. Ungar, J.A White, Agile user centered design: enter the design studio, case studies, Proceedings of the Conference on Human Factors in Computing Systems (CHI '2008), ACM Press, 2008, pp. 2167–2177.

[36] H. Sharp, H. Robinson, J. Segal, Integrating user centered design and software engineering: a role for extreme programming. Available: <https://pdfs.semanticscholar.org/3cb8/3f56899f78b9f9f1f0039a9fa3471d415c0f.pdf>, (Accessed 28 October 2018).

[37] J. Patton, Hitting the target: adding interaction design to agile software development, Proceedings of the Conference on Object Oriented Programming Systems Languages and Application (OOPSLA 2002), ACM press, 2002, pp. 1–7.

[38] J. Eklund, C. Levingston, Usability in agile development, Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.509.363&rep=rep1&type=pdf, (2008) , Accessed date: 28 October 2018.

[39] H. Obendorf, M. Finck, Scenario-based usability engineering techniques in agile development processes, case study, Proceedings of the Conference on Human factors in computing systems (CHI '08), 2 ACM Press, 2008, pp. 159–2166.

[40] A. Holzinger, M. Errath, G. Searle, B. Thumher, W. Slany, From extreme programming and usability engineering to extreme usability in software engineering

education (XP + UE- > XU), Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05), IEEE Press, 2005, pp. 169–172.
[41] S.E. Toulmin, R.R. Rieke, A. Janik, An Introduction to Reasoning, Macmillan, New York, 1984.
[42] P. Clarke, R.V. O'Connor, The situational factors that affect the software development process: towards a comprehensive reference framework, Inform Softw Technol 54 (5) (2012) 433–447.
[43] R. Wieringa, N. Maiden, N. Mead, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requirements Eng 11 (1) (2006) 102–107.
[44] T.S. da Silva, M.S. Silveira, C. de O. Melo, L.C. Parzianello, Understanding the UX designer's role within agile teams, in: A. Marcus (Ed.), Design, User Experience, Usability. Design Philosophy, Methods, and Tools, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 599–609.
[45] L. Miller, Case study of customer input for a successful product, Proceedings of the Agile Conference'05, IEEE Computer Society, 2005, pp. 225–234.
[46] G. Cockton, M. Lárusdóttir, P. Gregory, Å. Cajander, Integrating User-Centred Design in Agile Development, Human-Computer Interaction Series, 1, Springer International Publishing, 2016 276.
[47] A.B. Marques, R. Rodrigues, T. Conte, Systematic literature reviews in distributed software development: a tertiary study, 2012 IEEE Seventh International Conference on Global Software Engineering, 2012, pp. 134–143, , https://doi.org/10.1109/ICGSE.2012.29.
[48] M. Bano, D. Zowghi, N. Ikram, Systematic reviews in requirements engineering: a tertiary study, 2014 IEEE 4th Int. Work. Empir. Requir. Eng. Emp. 2014 - Proc, 2014, pp. 9–16, , https://doi.org/10.1109/EmpiRE.2014.6890110.

Karina Curcio has a Bachelor's degree in Informatics, Master's in Electrical Engineering at Federal University of Technology - Paraná. She is a PhD. candidate at Pontifical Catholic University of Paraná and her research interests are in software engineering, software quality, agile software development and innovation.

Rodolfo Santana has a Bachelor's degree in Informatics, Specialist in Software Engineering. He is a Master's candidate at Pontifical Catholic University of Paraná and his research interests are in software engineering, software product lines and open innovation.

Sheila Reinehr has a Bachelor's degree in Mechanical Engineering, Master's in Informatics and Doctorate in Engineering. Professor at Pontifical Catholic University of Paraná with experience in software engineering, software process improvement, agile software development, software quality, and software engineering education.

Andreia Malucelli has a Bachelor's degree in Informatics, Master's in Electrical Engineering and Doctorate in Electrical and Computing Engineering. Professor at Pontifical Catholic University of Paraná with experience in software engineering, artificial intelligence, organizational learning, ontologies and software engineering education.