Contents lists available at ScienceDirect

Information and Software Technology

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

Approaches to manage the user experience process in Agile software development: A systematic literature review

Andreas Hinderks [a,*], Francisco José Domínguez Mayo [a], Jörg Thomaschewski [b], María José Escalona [a]

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

ARTICLE INFO

Keywords:
User experience management
UX process
User experience
UX
Usability
HCI
Agile methods
Agile
Systematic literature review

ABSTRACT

Context: Software development companies use Agile methods to develop their products or services efficiently and in a goal-oriented way. But this alone is not enough to satisfy user demands today. It is much more important nowadays that a product or service should offer a great user experience — the user wants to have some positive user experience while interacting with the product or service.
Objective: An essential requirement is the integration of user experience methods in Agile software development. Based on this, the development of positive user experience must be managed. We understand management in general as a combination of a goal, a strategy, and resources. When applied to UX, user experience management consists of a UX goal, a UX strategy, and UX resources.
Method: We have conducted a systematic literature review (SLR) to analyse suitable approaches for managing user experience in the context of Agile software development.
Results: We have identified 49 relevant studies in this regard. After analysing the studies in detail, we have identified different primary approaches that can be deemed suitable for UX management. Additionally, we have identified several UX methods that are used in combination with the primary approaches.
Conclusions: However, we could not identify any approaches that directly address UX management. There is also no general definition or common understanding of UX management. To successfully implement UX management, it is important to know what UX management actually is and how to measure or determine successful UX management.

1. Introduction

Today's users expect a high level of satisfaction while interacting with a product. They expect to be able to use the product without any major effort to finish their tasks in a quick and efficient manner. Moreover, for a product to succeed, it is important to consider hedonic interaction qualities — i.e. those that are not directly target-oriented [1]. In summary, the user wants to have a positive user experience while interacting with any product or service.

A well-known definition of user experience is given in ISO 9241-210 [2]. Here user experience is defined as 'a person's perceptions and responses that result from the use or anticipated use of a product, system or service'. Therefore, user experience is viewed as a holistic concept that includes all types of emotional, cognitive, or physical reactions concerning the concrete or even only the assumed usage of a product formed before, during, and after use.

A different interpretation defines user experience as a set of distinct quality criteria [1] that includes the classical usability criteria or pragmatic qualities, such as efficiency, controllability, or learnability, and non-goal directed or hedonic quality criteria [3] like stimulation, novelty, or aesthetics [4]. This definition has the advantage that it splits the general notion of user experience into a number of quality criteria, thereby describing the distinct and relatively well-defined aspects of user experience.

Software development companies use Agile methods to develop products or services more efficiently. Agile methods (e.g. Scrum [5], Kanban [6], or Extreme Programming (XP) [7]) reduce the time taken to develop a product available in the market [8]. The iterative approach to developing software minimizes the risk of developing software that is not in line with what is needed in the market [9]. By performing

* Corresponding author.
E-mail addresses: andreas.hinderks@iwt2.org (A. Hinderks), fjdominguez@us.es (F.J. Domínguez Mayo), joerg.thomaschewski@hs-emden-leer.de (J. Thomaschewski), mjescalona@us.es (M.J. Escalona).

https://doi.org/10.1016/j.infsof.2022.106957
Received 20 October 2020; Received in revised form 6 September 2021; Accepted 19 May 2022
Available online 30 May 2022

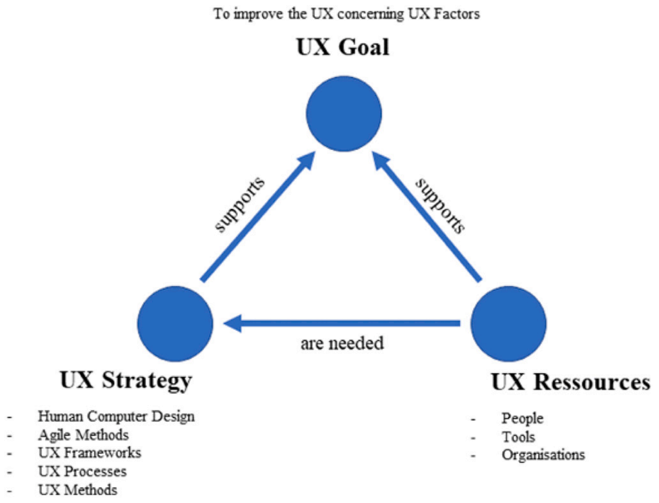

Fig. 1. User Experience Management based on McKeown [12].

retrospectives [5] at the end of an iteration, both product quality and Agile process quality can be improved.

To develop the best possible product with great user experience, it is essential to have the right management in place in terms of UX. To the best of our knowledge, there is no approved definition of UX management in the literature. There is also no common understanding of what UX management is or how to apply it (Section 2).

We generally understand management based on the explanations of Drucker [10] and Stone [11] — it is a combination of a goal, a strategy, and resources. When applied to UX, user experience management consists of a UX goal, a UX strategy, and UX resources (Fig. 1) based on the work of McKeown [12].

For example, a UX goal can be fixed to improve UX for a particular factor or quality criteria of UX. For this purpose, a UX strategy can be developed from different UX methods. For instance, to reach the UX goal, you can conduct a survey with a UX questionnaire, such as the User Experience Questionnaire [13] or the SUPR-Q [14], before and after the development. The UX strategy is that the results from the questionnaires after the development should be better than what they were before the development.

Both UX strategy and UX resources are necessary to achieve the UX goal. It should be known before the next development iteration, whose requirements positively supported the UX goal. In this way can the UX goal be achieved in a goal-oriented manner.

This paper reports the findings of a systematic literature review (SLR) in the field of approaches to manage the user experience process by focusing on Agile software development. This SLR will be addressed by the following research questions:

- RQ1: Which approaches are suitable for UX management in an agile context?
- RQ2: What conclusions can be deducted from the studies found?
- RQ3: How can user experience in Agile software development be planned and controlled for a product backlog item or a requirement before the development?
- RQ4: What retrospective proposals exist to improve the efficiency and effectiveness of the user experience process in terms of Agile software development?

This paper is structured as follows: Section 2 briefly summarizes the related work and presents gap analysis. Section 3 present the review method including research questions of this SLR, search strategy, selection process, quality assessment, and data extraction. Section 4 outlines the results and key findings of our study as well as the answers to our research questions. Section 5 discusses the meaning of the findings and the limitations of our study. The paper ends with Section 6, with conclusions and ideas for future work.

2. Background and related work

As already mentioned in the introduction, we did not find a definition of UX management nor a common understanding of the term in the literature. We searched for “user experience management” and similar terms in IEEEXpore, Science Direct, Scopus, Springer Link and ACM. The full search string we used was: (“user experience management”) OR (“manage user experience”) OR (“ux management”) OR (“manage ux”). In the end, we found five relevant paper. In these paper, there are various approaches or descriptions of UX management. The term UX management is often used without any explanation. We present the five paper in the next two paragraphs.

In the literature, the term UX management is used differently. Szóstek [15] used the term UX management in the context of team building and empowerment. This includes career planning and development, team management, and training of individual team members. Anderson et al. [16] used a similar approach — in addition to building a UX team, they proposed that a C-level executive focus on user experience is necessary for UX management to have any corporate influence at all. For the implementation of UX management, Anderson et al. [16] and Rosenberg [17], for example, offered various patterns that provide support at the levels of planning, decision, tactics, and conflict.

Another approach is the use of a UX maturity model. The advantage of using such a model is that it determines the current maturity level of an organization. Thus, its weaknesses can be identified. But the decisive factor is which dimensions are mapped in the UX maturity model. For example, the *Total User Experience Management (TUXM)* [18] model contains elements such as UX objectives, integrated design system, strategic communication, continual improvement, fact-based decision-making, and a T-type design team. The *Nielsen Corporate Usability Maturity Model* [19], on the other hand, comprises dimensions such as the developers’ attitude towards usability, the management’s attitude towards usability, the usability practitioner’s role, usability methods and techniques, and strategic usability. At first glance, it is noticeable that the TUXM model contains the dimension called UX objectives which is not present in Nielsen’s model. Conversely, the Nielsen model is more focused on practical implementation. The testing of a suitable UX maturity model should be carried out before deployment and tailored to the needs of the organization [19].

To the best of our knowledge, we did not find any paper in the literature that considers both managing UX process and Agile software development. But we found papers that analyse the integration of UX or similar methods and Agile software development. Therefore, the next section summarizes an overview of SLRs regarding the integration of UX and Agile software development.

2.1. Summary of related literature reviews

In the literature, there are many reviews that investigate the integration of HCI and Agile methods. The term ‘Agile methods’ is used in the same way by all SLRs. However, there are differences in the processes or methods from the HCI area being used or integrated. The range of methods includes classic usability engineering, user-centred design (UCD) or human-centred design (HCD) [2], and UX methods in general, as well as design thinking.

The next paragraphs briefly summarize the SLRs found on the basis of our search results (Section 4.1). In Fig. 2, the SLRs are arranged in chronological order on a timeline.

The 2010 SLR by Bruun [20] investigated whether developers are trained in usability engineering so that they can apply usability engineering methods themselves. One of the main results of the SLR is the following finding: usability engineering is mainly published with

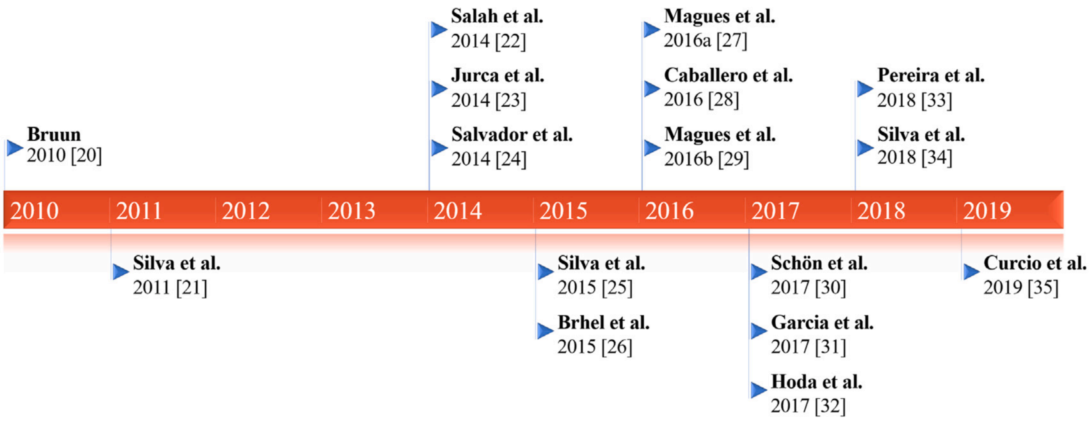

Fig. 2. Systematic Reviews in a Chronological Order.

a university or laboratory focus. Only a small part was dedicated to training the developers of usability methods. In the end, only one study could be identified that covered the essential aspects, namely user-based methods, training costs, focus on organizational contexts, and practitioners.

In 2011, Silva et al. [21] conducted an SLR on the integration of Agile methods and user-centred design. The authors analysed how usability problems are handled in Agile projects. The authors identified the following key aspects that play an essential role in integration: little upfront design, prototyping, user stories, user testing, inspection evaluation, and one sprint ahead.

In 2014, Salah et al. [22] analysed the current state of Agile and User-Centred Design Integration (AUCDI). Their analysis should identify the factors for the integration of Agile method and user-centred design. Besides, the authors examined the challenges and key aspects to ensure a successful integration. The identified key aspects are lack of allocated time for upfront activities, difficulty of modularization, optimizing the work between developers and UCD practitioners, performing usability testing, and lack of documentation.

In 2014, the SLR of Jurca et al. [23] analysed the literature to derive findings and recommendations for the integration of Agile and UX (Agile-UX). One finding is that Agile-UX methods are not anchored in companies and therefore do not receive the necessary support. Besides, UX designers are reworked and not part of the development team, but they are responsible for several development teams. This has been shown to reduce the efficiency and effectiveness of UX designers.

Salvador et al. [24] conducted in 2014 an SLR with the focus on which usability methods are used in Agile methods and when. The most commonly used usability methods include: fast prototyping, individual inquiry, formal tests, and heuristic evaluations. These methods are used about 50% during the implementation phase and 40% during the design phase. Only 10% of the usability methods used are implemented automatically.

In 2015, Silva et al. [25] performed a systematic mapping and analysed publications from significant Agile and HCI conferences. The objective was to answer the research questions on how Agile UCD is understood and which techniques are used in Agile UCD. Agile UCD is generally understood in the same way. It applies equally to the phases research, design, prototype, and evaluate. The most common technique is the implementation of usability test on lightweight prototypes.

In 2015, Brhel et al. [26] published an SLR by stating the principles of user-centred Agile software development (UCASD). The authors aimed to assess the current state of the art regarding the integration of Agile software development and user-centred design. Using a coding system, the authors extracted five derived principles: separate product discovery and product creation, iterative and incremental design and development, parallel interwoven creation tracks, continuous stakeholder involvement, and artefact-mediated communication.

In 2016, Magües et al. [27] conducted a systematic mapping study (SMS) in order to determine the current status of the integration of usability techniques in Agile processes. To that end, 31 studies were analysed and the usability techniques used were assigned to the development phase (requirements engineering, design, and evaluation). The most frequently used usability techniques for requirements engineering are ‘personas’; for design, ‘low-fi prototyping’; and for evaluation, ‘usability expert evaluations’.

Likewise in 2016, Caballero et al. [28] conducted a literature review to investigate the extent to which Agile teams integrated UCD methods in their Agile software development process. One result was that the most frequently used Agile methods are Scrum and XP. The three main UCD methods, which represent 70% of the methods used, are prototypes, user stories, and usability testing.

Also in 2016, the results from the SLR by Magües et al. [29] were further analysed in a mapping study by Magües et al. [27]. According to Brhel et al. [26], the selected studies were classified into the categories ‘process integration’ (48%), ‘practice integration’ (19%), ‘team integration’ (17%), and ‘technology’ (4%). The remaining papers could not be directly allocated. In conclusion, the authors concluded that there are no formalized suggestions for integrating usability techniques in Agile software development.

In 2017, the systematic literature review by Schön et al. [30] focused on approaches and methods for involving stakeholders in the process of Agile requirements engineering. A total of 27 papers were analysed. The most important result: there is no common understanding of the user perspective in Agile software development. However, four methods (Human-Centred Design, Design Thinking, Contextual Inquiry, and participatory design) were identified that integrate knowledge of user needs in Agile software development.

Garcia et al. [31] conducted a systematic mapping study in 2017. The purpose was to investigate artefacts used in communication between Agile methods and User-Centred Design. A total of 20 artefacts were identified and examined, such as prototype, user story, scenario, sketch, persona, and card like the design card or the task-case card. During the development iteration, about 56% of the artefacts are used. The rest are used during the discovery or planning phase.

In a meta-study in 2017, Hoda et al. [32] examined SLRs that treat Agile software development. A restriction to HCI was not made. The aim was to identify which developments in Agile software development can be recognized by the SLRs investigated. One finding is that the significant integration of established domains such as usability, CMMI, and global software engineering can be recognized. Usability is the second-most common (18%) integrated domain.

To evaluate how the Design Thinking approach is used in conjunction with Agile software development methods, Pereira & Russo [33] in 2018 used a systematic literature review. In total, 29 articles were collected, categorized, and reviewed. The results show that most integrated models are applied throughout the software lifecycle. In most cases, the design thinking approach of the International Organization for Standardization (ISO) was integrated in Scrum as an Agile method.

In 2018, Silva et al. [34] analysed the results obtained by Brhel's SLR [26] concerning the state-of-the-art integration of Agile methods and the user experience design. The outcome from the respective publications was divided into three dimensions: process and practice, people and social, and technology and artefacts. As a result, the individual outcomes were arranged on a timeline so that the chronological sequence of the publications was visualized. The authors stated in their analysis that solutions are already being offered for the dimensions process and practice and people and social concerning integration. Finally, the authors concluded that technology and artefacts are still missing to integrate Agile methods and user experience design with Agile UXD.

In a meta-study conducted in 2019, Curcio et al. [35] examined SLRs concerning Agile methods and usability. The fundamental question concerned how usability methods could be integrated in Agile software development. It was found that there are different levels of integration — process, practices, team, and technology integration. The biggest challenges are issues related to tests, time, work balance, modularization, feedback, prioritization, and documentation. Another important finding is that the type of integration that has evolved from two independent teams (parallel track) to one team during the search period.

In total, we presented 16 SLRs for the integration of UX in Agile software development. The number of SLRs indicates that the integration has met the scientific interest. Besides, the SLRs show that everyone has a different focus on integration. Finally, the SLRs presented here show positive progress in the integration of UX in Agile software development.

2.2. Gap analysis

In a further step, we investigated the research questions of SLRs. We assigned each research question to the category UX strategy, UX resources, or UX goal depending on the objective of the research questions. The categorization was done based only on the purposes of the research questions. The results of the research questions were not further investigated. A total of 47 research questions from the 16 SLRs were examined. Twenty-nine research questions were assigned to the category UX strategy, 7 UX resources, and 0 UX goal. The remaining 11 research questions could not be assigned to any of the categories.

The results of the research show a focus on UX strategy. This is remarkable in that a UX strategy should always start with a UX goal as a prerequisite or objective. Only if a UX goal has been defined, a corresponding UX strategy can be selected. Every UX strategy indeed leads to a UX goal, but the definition of this goal is undefined and therefore, not manageable. From our point of view, all three categories have to be covered if managing UX is needed.

The studies from the related literature review deal in different levels with the integration of UX or HCI in Agile software development. Garcia et al. [31] and Bruun [20] for example provide approaches for measuring the success of the applied approaches in terms of improving UX. However, we found that the next step is to focus on the strategic pursuit of user experience improvement. It was assumed that the integration of UX methods in Agile software development improves the user experience of the product, but this cannot be measured. However, to determine whether a goal has been achieved, the previous, expected, and post-implementation state should be measured [12]. This is the only way to determine whether the UX strategy and UX resources used have achieved the UX goal (Fig. 3). In other words: Has desired UX been achieved?

We did not find a systematic literature review which investigates user experience management by focusing on Agile software development directly. The research questions tend to focus on UX strategies and perhaps UX resources, but not on UX goals. For this purpose, we conducted this SLR.

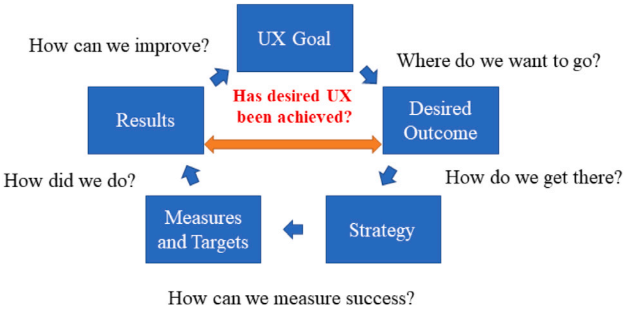

Fig. 3. Desired Outcome vs. Results based on the Strategic Planning Cycle [12].

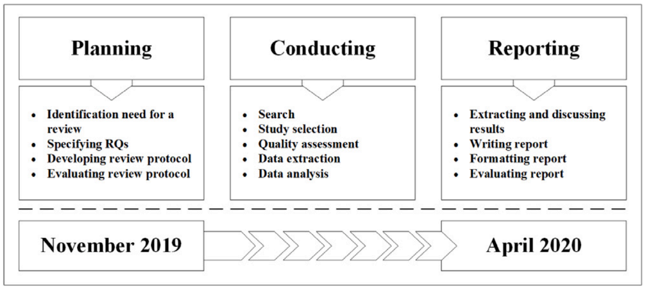

Fig. 4. Phases of an SLR.

3. Research methodology

Appropriate guidelines have been followed for conducting a systematic review, particularly the guidelines for SLR in software engineering by [36]. According to these guidelines, our SLR consists of three main phases. Fig. 4 shows the most important stages of each phase.

Owing to the high number of retrieved studies, we used the SLR Tool [37] and the software Citavi in order to manage information obtained in an efficient manner. We used the SLR tool when performing the SLR (managing the paper, developing the review protocol, documenting the search, and conducting quality assessment). In our literature database managed with Citavi, we imported the result of the SLR to use the management and citation functions.

3.1. Objectives and research questions

In the beginning, we did informal research on UX management or related terms. We conducted the informal research with Science Direct, Springer Link, IEEEXplore, Scopus, and ACM with the keyword 'user experience management' and variations of it.

The result was presented at the beginning of Section 2. However, during the research, we also found that the term 'UX management' is neither sufficiently defined nor explained in the literature. Further, we found through GAP analysis (Section 2.2) that there was a research gap in the goal, strategy, and resources concerning UX management. Besides, our informal research revealed that the number of papers found was too small and their content too widely scattered. However, we found approaches that allow UX management, as described in Section 2.2. This is the basis for our research questions.

RQ1: Which approaches are suitable for UX management in an agile context? This question aims to identify approaches that can potentially be used for UX management. We did not expect that the approaches that had been found could be used explicitly for UX management. Otherwise, we could have already identified approaches in the literature search (Section 2). Our analysis was intended to list approaches that were generally successful or had a high acceptance concerning UX methods in Agile software development.

Table 1
Keywords used for search.

Category	Keywords
Agile	Agile, Kanban, Scrum, Lean, Extreme Programming, Design Thinking
User Experience	User Experience, UX, Usability, HCD, HCI, HMI, UCD

RQ2: What conclusions can be deduced from the studies found? Concerning these research questions, we wanted to find out whether, in addition to approaches, other findings on UX management could also be derived from the studies. Not every study contains an approach that can be used directly for UX management. We instead assumed that studies would be found which described the integration of user experience methods and Agile software development. These aspects also need to be considered and analysed.

RQ3: How can user experience in Agile software development be planned and controlled for a product-backlog item or a requirement before development? Management also implies a goal — what is to be achieved so that the necessary strategy and resources can be selected? The third research question aims to identify approaches that can be used to estimate UX product-backlog items before development. The result of the estimation is to figure out the potential UX that can be reach if the product-backlog item will be developed. In this way, the estimation of UX can be used to determine where there is potential to achieve a potential UX goal. Further, the question remains as to what extent the estimated UX can be expressed in the form of product-backlog items or requirements.

RQ4: What retrospective proposals exist to improve the efficiency and effectiveness of the user experience process in terms of Agile software development?

In terms of these research questions, our goal was to identify proposals for improving the UX process. We had to consider the fact that Agile software development was usually iterative. This means that after each iteration, there is the possibility of improving the UX process.

3.2. Search strategy and data sources

Based on the research questions and research objectives, we developed a search strategy. This strategy contains the search string, the search space, and the process to select the relevant papers.

The first step is to create a set of keywords. Since UX management has not been sufficiently covered in the literature (Section 2.2), the set of keywords consists of *Agile* and *user experience* as far as related terms are included. In practice, it has been shown that *Agile* is often not directly addressed, but rather *Kanban*, *Scrum* etc. Agile frameworks, like *Scrum*, or agile methods, like *Kanban*, are often used as a keyword in combination with Agile Methods. For this reason, we have included agile frameworks and methods in the search string. We also included the term *design thinking* because our experience has shown that useful publications have also been found with this term.

In the second step, we extended the keywords by alternative spellings and synonyms. These were extracted from the previously analysed literature (Section 2). Finally, we consolidated and optimized the list of keywords. The final list of keywords is shown in Table 1.

The set of keywords was then transferred to a search string in the next step. This is as follows:

(agile OR kanban OR scrum OR lean OR “extreme programming” OR “design thinking”)

AND

(“user experience” OR ux OR usability OR hcd OR hci OR hmi OR ucd)

Table 2
Search space with specification of search strategy (TAK = Title, Abstract, and Keywords) and number of paper.

Library	Search strategy	Number
IEEEXplore	Full Text	863
Science Direct	TAK	61
SCOPUS	TAK	1, 308
SpringerLink	Full Text	3, 874
ACM	TAK	26

This search string was adapted to the syntax of the respective search spaces as these had partially differed. The actual logic, however, had not been changed.

The search space included digital libraries, journals, and conference proceedings. A complete list of the search space is shown in Table 2. The search was conducted at all search spaces in January 2020.

Without any restriction – i.e. plain full-text search of the search engine – N_{P0} = 44,637 (Fig. 5) papers were found.

It should be noted that IEEEXplore and Springer Link had problems restricting the search to title, abstract, and keywords. Both did not offer the possibility to search for title, abstract, and keywords together. The conversation with the support of the respective providers has also led to no result. These problems were partially resolved by the owner of the search engines, but they led us to a slightly different strategy. Wherever possible, we downloaded the paper and put the abstract and keywords into plain text. From 4496 paper, we were able to extract the abstract and keywords from 2733 paper. Finally, we conducted an own search limited to title, abstract, and keywords using the SLR Tool [37] on all N_{P1} = 4496 paper. The result was that we had to check N_{P1} = 4496 (Fig. 5) paper initially.

By conducting the internal search function of the SLR Tool [37], we could reduce the result by searching only on title, abstract, and keywords so that the amount of paper to be examined was N_{P2} = 1253 (Fig. 5). This data set was further examined by us, as described in Section 3.3.

3.3. Study selection

In the previous section, it was described how the number of papers was limited by the search criteria (Step N_{P2}). In a further step, we reduced the number of papers from N_{P2} = 1253 to N_{P3} = 196 by scanning the title. We only included those papers that were interesting and valuable for our SLR in terms of their title. The title should be recognized that the paper is mainly about ‘agile’ and ‘user experience’. If the title indicated that the paper only applied agile and UX methods, the paper was excluded. In the following step, we reduced the number of papers to N_{P4} = 110 by reading the abstract. We applied the same criteria we used one step ago. All the decisions are traceably logged by the SLR tool.

In each step of the reduction, a set of selection criteria were applied. These which are divided into inclusion and exclusion criteria.

The inclusion criteria were: papers written in English; papers under peer-reviewed papers; and papers presenting approaches to integrate user experience methods (or similar) in Agile development processes.

Exclusion criteria were: no full books; papers whose full text were not available; papers only presenting lessons learned, ideas, guidelines or recommendations; papers introducing a panel talk or a workshop at a conference; papers with results that had already been published; papers that were not focused on Agile development; papers introducing tools whose underlying methodology was not comprehensibly described (black box).

After the study selection, we performed a snowballing process according to Wohlin [38]. We applied forward snowballing (search in papers that cite the paper) and backward snowballing (search in the reference list of the paper). Snowballing has the advantage that we

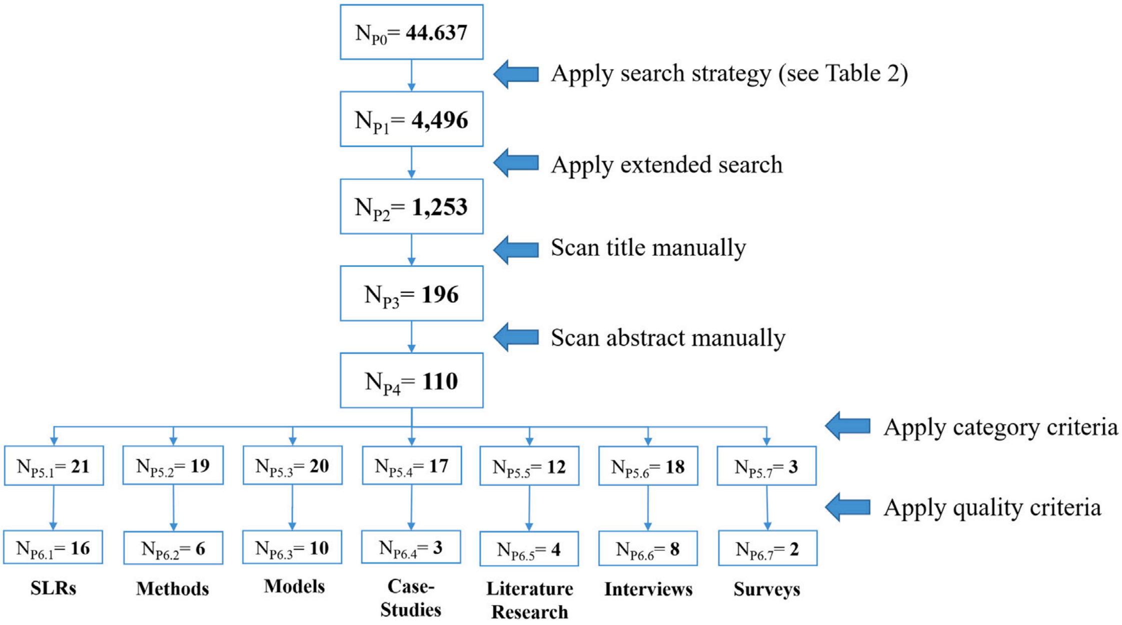

Fig. 5. Search Process comprising Phases and Inclusive Forward and Backward Snowballing.

Table 3
Quality checklist for empirical studies.

Item	Assessment criteria	Score	Description
QA1	Was more than one study conducted?	−1	Only one study was conducted
		0	Two studies were conducted
		1	More than two studies were conducted
QA2	Was the target group selected randomly?	−1	No, randomized group of participants
		1	Yes, randomized group of participants
QA3	Is the data analysis process appropriate?	−1	No analysis has been taken
		0	One statistical analysis has been taken
		1	The dataset is well analysed
QA4	Is the result of the statistical analysis appropriate?	−1	No or poor results
		0	The results are okay
		1	The results are good enough

can identify additional papers important to the SLR that were not identified via the SLR method itself in addition to the systematic search. The additionally found papers (N = 29) were inserted in step N_{P4} (Fig. 5). The numbers shown in Fig. 5 include the papers added by the snowballing. In total, seven additional papers for data extraction were added at the end.

3.4. Quality assessment

The papers selected in the previous section (N_{P4} = 110) were evaluated with a quality assessment. We developed a checklist (Table 3) based on the recommendations of Kitchenham and Charters [36] (Table 3) to evaluate case studies, literature research, interviews, and surveys. Methods and models were excluded because they are qualitative studies. To better classify the papers, we classified the papers according to case studies, literature research, interviews, surveys, methods and models. This classification is based on our own created system. We manually reviewed these methods and models by reading and evaluating the paper. The evaluation was based on the basic orientation of the study and whether it is suitable for our SLR.

The overall aim was to identify studies of low quality and then exclude them from our study.

In the end, every paper was rated with a sum of the individual result. We decided to include those articles with a score greater than or equal to 1.

The SLRs determined were checked to see whether the SLR was carried out in a traceable manner. Also, we checked whether the SLR was performed according to a standard published in the literature. The SLRs were then reduced to $N_{P6.1}$ = 16.

Models and methods were generally reviewed for evaluation or validation. The aim was to determine whether the method or model was generally successfully applied in a study. After validation, we reduced the methods to $N_{P6.2}$ = 6 and models to $N_{P6.3}$ = 10.

After the user of the quality assessment, we reduced the case studies to $N_{P6.4}$ = 3, literature research $N_{P6.5}$ = 4, interviews $N_{P6.6}$ = 8, and surveys to $N_{P6.7}$ = 2.

In the end, we included 49 papers out of 110 papers in our SLR study.

3.5. Data extraction and analysis

According to Kitchenham's and Charters's guidelines [36], a form for data extraction was set up. We used the SLR Tool [37] in order to gather summaries, etc. The SLR Tool also supported the data extraction with regard to defined attributes from the protocol:

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

Table 4
Distribution according to research methods and year.

Research method	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Total	
SLR				1	1			3	2	3	3	2	1	16	32%
Methods				1				2		2			1	6	12%
Models	1	2		1	1	1		2		1	1			10	21%
Case studies						1			1	1				3	6%
Literature research							1	2			1			4	8%
Interviews	1	1						1	2	1	1		1	8	17%
Surveys						2								2	4%
Total	**2**	**3**	**0**	**3**	**2**	**4**	**1**	**10**	**5**	**8**	**6**	**2**	**3**	**49**	**100%**

In addition to the automatically extracted data, we have determined the following attributes manually:

- Paper category: e.g. SLRs, models, methods, case studies, literature research, interviews, and surveys
- Used UX methods: e.g. personas, prototypes, usability evaluations
- Agile methodology: e.g. Scrum, XP, Agile in general
- UX process integration: Parallel track or one track
- Development phase of usage: before, during, or after
- Short summary
- Results and contributions
- Personal assessment

In some cases, it was not possible to fill every attribute. In this case, we filled the attribute with ‘not specified’. In the last step, we checked the content of all studies and assigned them to particular research questions. The aim was to have a list of studies per research question so that we can use to answer them. The research questions are answered in the next section.

4. Results

In our work, we have selected 49 relevant studies. In this section, we will present the studies individually or in total, if necessary, to answer the research questions. The first part of this chapter gives an overview of the selected studies. In the second part, the individual research questions will be answered based on the studies.

4.1. Summary of the studies

Our search was limited to ‘Agile’ and ‘UX’ (and similar terms). An explicit restriction to UX management or similar was not made. This is because already in our first literature review it was found that the search result would not be sufficient to answer the research questions (Section 2). For this reason, the studies included in our SLR cover a broad spectrum. Incidentally, all the studies have been examined for their applicability in the field of UX management.

In journals, 15 (31%) of the included studies and 34 (69%) in conference proceedings were published. In Table 4, all 49 studies analysed in this SLR are grouped by research method and year. 15 (31%) of the studies were published between 2007 and 2013 and 34 (69%) between 2014 and 2019. It should also be noted that our research was conducted in January 2020, so there may still be late publications that are not included in this SLR.

All the 49 included studies were assigned different underlying research methods (Table 4). Of the studies included, 16 (32%) are *structured literature reviews (SLR)*. These SLRs have not been used to answer the research questions as they themselves answer research questions that differ from ours. However, they have been presented in section Related Work (Section 2.1) and form the basis of the gap analysis (Section 2.2).

Further, six (12%) of the studies included in this study can be assigned to the category *Methods* [39–44]. Methods are to be understood rather generally in this category. Usually, a procedure is described, how a problem was solved. A total of eight different methods were presented in these studies. Of the eight methods, four known methods were newly combined (Tool for A/B test [44], Checklist for a possible maturity model [43], Personas [40], Nielsen’s heuristics [42]), three new methods were presented (Usability Goals Achievement Metric (UGAM) [39], Index of Integration (IoI) [39], Web business process refactoring (WBPR) [41]), and one method was supplemented (UserX Story [42]).

The 10 (21%) *Models* [45–54] included can be divided into four frameworks, three conceptual models, two processes, and one lifecycle. For the sake of simplicity, they have been assigned to the research method models, since in some cases even the author of the study has not specified the research method.

All three (6%) *Case Studies* [55–57], four (8%) *Literature Research* [19,58–60], eight (17%) *Interviews* [55,61–67], and two (4%) *Surveys* [68,69] have been conducted in economic enterprises in such a way that the results and conclusions from each study are very practical. The Case Studies can be summarized as ‘classic case studies’. As *Literature Research*, we have categorized studies that answer research questions based on the literature. In this case, the difference between an *SLR* and *Literature Research* depends on the used method to conduct study. The *Interviews* were partly conducted as semi-structured interviews. Finally, the included Surveys were conducted in the classical way with a self-developed questionnaire.

Of the total of 49 studies, 16 *SLRs* are included, which we discussed in Section 2. Of the 33 remaining studies, 38 individual studies are included and analysed in this section. Two are *Multi-Case Studies* with four and two *Case Studies*, respectively, and two *Methods* are presented in one study. For this reason, 38 individual studies are selected and presented in our paper. The corresponding studies with several individual studies are numbered accordingly in round brackets. These are described in the next five sections. We further analyse the approaches in the 38 individual studies. We use the term approach as a generic term for a method, a model, a case study, literature research, an interview, or a survey.

4.1.1. User experience or usability

The 38 individual studies were examined as to whether each of them used the concept of *usability* or *user experience*. In some cases, it was not possible to assign the concept, as both concepts were used. In this case, the concept that was the most present in the individual study had been selected. Of the 38 individual studies, 16 (42%) use the concept usability and 22 (58%) the concept of user experience.

4.1.2. Used Agile method

We also examined the 38 individual studies according to the Agile method used. A total of 21 (55%) of the studies have not been assigned to a specific method. This is because the authors of the paper do not address explicit Agile methods. This means that it could be used for any Agile methods. The rest is divided between Scrum 11 (29%) and Extreme Programming 6 (16%).

4.1.3. Type of integration of UXin Agile methods

We also examined the 38 individual study to determine whether UX methods were used within the development team or outside as an additional parallel track/iteration. Of the individual studies examined, 11 (29%) integrated UX methods in the development team. The UX methods are usually used by UX professionals as well as developers. The eight (21%) individual studies, on the other hand, which use a UX team with its organization, in addition to the development team, are different. With an additional UX team, this team does not necessarily have to work exclusively with the development team, but can also work for several teams. The remaining 19 (50%) individual studies are not further specified for integration. One reason for this is that the individual study condenses results in such a way that the type of integration can no longer be deduced from them. Another reason may be that individual UX professionals only work with the development team and therefore do not constitute a team in their own right.

4.1.4. Proposed approaches

In the next step, we extracted the proposed approaches from the 38 studies. In total, we were able to extract 18 unique primary (Table 5) approaches from 24 individual studies out of the 38 studies. We analysed these in a further step. The results are presented in Sections 4.1.5 and 4.2.1.

4.1.5. Time period of use

In a further step, all 18 approaches were analysed with their temporal applicability in development. The aim was to examine each approach about the phase of development in which this approach is to be or was used. The breakdown was made according to before, during, or after development. If an approach can be used in several phases of development, it was also assigned to those phases. In total, 21 (88%) approaches can be used before development, while 15 (63%) approaches can be used during development and 13 (54%) after development.

In the next paragraphs, we answer the individual research questions from Section 1 using the 38 individual studies and the 18 primary approaches extracted from the 38 individual studies. All studies, except for the SLRs determined, serve as a basis.

4.2. (RQ1) which approaches are suitable for UX management in an agile context?

The first research questions seek to answer which of the 18 primary approaches extracted from the studies are suitable for the management of UX. One criterion for determining when an approach is suitable is when the presented approach has been successfully applied within Agile software development. Thus, the approach presented in the study is suitable for UX management for the first time. The quality assessment (Section 3.4) already ensures that the approach has been sufficiently evaluated or validated.

Table 5 lists all the primary approaches that were mainly presented or used in the individual studies. This means that in addition to the primary approaches, additional UX methods have been used. The main focus for answering RQ1 is on the primary approaches. Additionally used UX methods can support the primary approach, but they would not bring the desired success if used in isolation. The additional UX methods (Table 6) that were used in combination with the primary approaches are described in Section 4.2.2.

4.2.1. Primary approaches presented in studies

In the following sections, we present the highlights found in the included studies. In Table 5, we list all 18 primary approaches presented in the studies (Section 4.1.4). These approaches are potentially suitable for managing UX processes because they integrate UX methods into agile software development.

The most frequently identified approach is *'Upfront UCD Design'*. A second UCD team will be added to the actual development team. This team works in a parallel track (Section 4.1.3), always one iteration ahead of the development iteration. In this parallel track, prototypes [55,61,70] are usually created in various forms. These prototypes are then handed over to the development team and developed in the next iteration of iterations. Silva et al. chose a similar approach [50] — their framework defines the tasks of the individual teams within the parallel track concerning the user experience.

The second-most frequently used method or approach is *'Communication/Collaboration'* — a simple but successful approach. The approach is implemented in various ways. Ferreira et al. [55], for example, chose four different approaches in their study. The UI designers and developers worked together and constantly exchanged in this study. The UI designer usually developed a UI prototype based on a specification or a user story. This was then discussed together with the developers and then implemented by the development team. The result was that the developers had a much better understanding of the goal of the specification or user story. Ferreira et al. [70] chose a similar approach in another study.

Øvad et al. [56] and Øvad and Larsen [71] followed a different approach. Both approaches aim to teach the developers selected usability and UX methods so that they can use them independently in their development work. The approaches include *'A/B Testing'* [56,71] as well as *'Focus Workshops'* [71] and *'Contextual Interviews'* [71].

The *'Cruiser Lifecycle'* developed by Memmel et al. [45] aimed at integrating the methods of human–computer interaction (HCI) in Agile software development. The lifecycle mainly consists of three phases. The result of the first phase (Initial Requirements Up-Front) is a collection of artefacts like use case diagrams, scenarios, and prototypes. These are then processed further in the second phase (Initial Conceptual Phase). In the second phase, the release plan, system metaphor, UI design, and UI specification are developed. In the third phase (Construction & Test Phase), the product is developed and then its usability is evaluated. If new requirements are to be implemented, work can begin again from Phase 1. Xiong and Wang [48] and Humayoun et al. developed similar phase models [49]. All of them are based on the fact that there is a certain number of phases in which certain HCI or UX tasks are to be completed.

Singh [47] extended Scrum to *'U-Scrum'* by adding another role called *'Usability Product Owner'*. This is to ensure that the usability represented by the newly created role is already considered when requirements are created. Both the Product Owner (PO) and the Usability Product Owner (UPO) work on the same level, but with different focuses. The PO corresponds to the defined role in the Scrum Method [5]. In contrast, the UPO has to defining a user experience vision. This should be considered in the requirements analysis. Finally, the UPO is responsible for the usability design and for creating the requirements together with the PO.

Wolkerstorfer et al. [46] integrated UCD methods within Extreme Programming (XP) to an *'Agile Usability Process'*. Before the first iteration of development, conducting user studies, personas, and usability tests are necessary. During the development, usability expert evaluations of the product increment are performed simultaneously with the unit tests established in XP. These provide direct feedback to the development as well as work for the next iteration.

Joshi at al. [39] created the *'Usability Goals Achievement Metric (UGAM)'*. We will present this approach in Section 4.4 in more detail. The second metric presented in this paper is *'Index of Integration (IoI)'* [39]. This metric represents, on a scale from 0 to 100, the level of integration of HCI activities in software development activities. With both UGAM and IoI, the development team can know how good their developed product regarding UX happens to be.

'Web business process refactoring' (WBPR), developed by Distante et al. [41], is a framework to capture the usability improvements of business process web applications. A WBPR includes the following

Table 5
18 Primary approaches presented in studies.

No	Primary approach	Studies	Total
1	Upfront UCD design	[55](3), [55](4), [61], [70](1)	4
2	Communication/Collaboration	[55](1), [55](2), [70](2)	3
3	Teaching UX Methods	[56], [71]	2
4	Cruiser Lifecycle	[45]	1
5	U-Scrum	[47]	1
6	Agile Usability Process	[46]	1
7	Inter-Combined Model	[48]	1
8	Usability Goals Achievement Metric (UGAM)	[39]	1
9	Index of Integration (IoT)	[39]	1
10	Three-Fold Integration Framework	[49]	1
11	UXD & AD Framework	[50]	1
12	Agile Usability Model	[52]	1
13	Agile UX Model	[51]	1
14	Web Business Process Refactoring (WBPR)	[41]	1
15	UserX Story	[42]	1
16	Checklist for User-Centeredness of Agile Processes	[43]	1
17	CSWR Framework	[44]	1
18	SIBAP	[53]	1
		Total	**24**

Table 6
Methods used in combination with primary approaches.

No	Method	Studies	Total
1	Prototyping (Low/High)	[45], [61], [48], [63], [55](2), [55](3), [55](4)	7
2	Personas	[61], [46], [48], [40], [42], [54]	6
3	Task/Usage Scenarios	[45], [48], [54]	3
4	Acceptance Test	[55](2), [55](3)	2
5	Focus Groups	[45], [71]	2
6	Expert Reviews	[45], [46]	2
7	UX Questionnaires	[45], [48]	2
8	Contextual inquiry	[61], [71]	2
9	Usability Testing	[61], [63]	2
10	User Evaluation	[46], [54]	2
11	Interviews	[48], [54]	2
12	A/B Testing	[56], [71]	2
13	Card Sorting	[45]	1
14	Brainstorming	[45]	1
15	Usability Inspection	[45]	1
16	FlexREQ	[52]	1
		Total	**38**

criteria: intent (usability qualities such as effectiveness, efficiency, and satisfaction), bad smell (an indicator of the lack of usability), motivation (description of the problem), and examples (describes the application). For each of these criteria, the authors provide instructions for creating WBPRs within the framework.

Navarro et al. [53] developed the *'Script-Based Aspect-Oriented GUI Prototyping (SIBAP)'* framework. The objective of the development was to create prototypes using a scripting language that can be used by both designers and developers. The prototype is the common artefact of designers and developers. The designers use the framework to create a prototype, which is then implemented by the developers. Thus, there is no media break between the prototyping of the designers and the development.

In summary, all the approaches presented in this section aim to promote a collaboration between UX professionals and developers. Some approaches directly promote communication, while others try to promote the exchange via a process or framework.

4.2.2. Additional UX methods included in primary approaches

UX methods are used in various ways. Usually, UX methods are used in addition to a process or a framework. Table 6 lists all the UX methods that were additionally used to the 24 primary approaches in the studies.

The two most frequently used methods are *Prototyping* and *Personas*. Prototyping and personas can be used as artefacts for the communication between UI designers and developers. The UI designers either develop a prototype together with the developers or work on it before the actual development. Personas, on the other hand, are usually used permanently. Various methods are used to determine the requirements — these are *task/usage scenarios, focus groups, contextual inquiry, user evaluation, interviews, A/B testing, card sorting, brainstorming,* and *FlexREQ*. The following methods are used to measure and evaluate the user experience: acceptance test, expert reviews, UX questionnaires, usability testing, and usability inspection.

The UX methods listed in Table 6 were identified mainly by Jia et al. [68] as UX methods that serve in connection with the integration of UX in Agile software development. According to the list of Jia et al. [68] these are workshops, prototyping, interviews, scenarios, personas, field studies, usability goals, usability evaluation, questionnaires, and heuristic evaluation. What is interesting about the study, however, is the difference between the frequency of use and the evaluation of each individual method. The *Personas* method is rated quite positively, but 38% of the participants consulted it only once a year. In contrast, over 60% of the users use *Prototypes* at least two or three times a month.

4.3. (RQ2) what conclusions can be deducted from the studies found?

This question aims to identify findings that are useful in the context of UX management. To answer this research question, we analysed all 38 studies.

Both frameworks – user-centred design (UCD) and Agile methods – were developed independent of each other. Some studies have investigated to what extent the two frameworks can be integrated. The fundamental difference between the two frameworks is the focus. Agile methods focus on code production and the work of the developers, whereas UCD focuses on the interface and the work of UX experts [58,60,66]. Agile methods do not support per se UCD methods, but they do not prevent them [60]. Communication between UX professionals and developers is one possible solution [60]. Additionally, a sprint zero can be performed to define the UX vision [66]. Another positive effect is that user research must be carried out before the actual development [66].

For both the UX professional and the developer, it is challenging to create a UX vision for a new product [62]. This is also confirmed by the study of Hokkanen et al. [64], who developed a list of UX qualities for new products. Besides, the study by Kuusinen [62] stated that too little time is spent on UX work and that individual disciplines could work better together.

There are several UX methods for being used within Agile software development (Section 4.2.2). Kuusinen [72] identified UX-related tasks in her study that can also be performed by a developer. These are tasks and not complete methods. For example, developers can clarify the user requirements. Developers can review UI designs or create their own UI designs. However, developers should be involved in the design decision so that the acceptance increases [72]. A retrospective view held by UX professionals and developers can improve the collaboration [72].

Various studies have also found that UX and Agile methods are not necessarily mutually exclusive [58,60,66]. Both UX and Agile methods are iterative; they support feedback; and they are multidisciplinary [58]. However, how successfully UX methods are integrated and used in Agile methods considerably depends on the team itself [59].

The management does not consider UX to be part of the business strategy [69]. This means that while the management considers it essential to use UX methods to increase UX, it is not really part of the strategy [65]. One reason for this may be that UX knowledge is not yet firmly anchored in the management so as to become part of the company's strategy. UX knowledge is often expert knowledge and therefore not understandable or applicable for everyone [65].

4.4. (RQ3) how can user experience in agile software development be planned and controlled for a product backlog item or a requirement before development?

To answer the third research question, we examined all the approaches from Table 6 that can be used before development (Section 4.1.5). Two approaches [39,42] were identified, but neither of them can meet the requirements of the research questions. However, since both the approaches are potentially suitable with limitations or extensions, we present them in the next sections.

Choma et al. [42] extended or supplemented the grammar of a user story with user experience aspects and usability requirements. New or replaced components of a *UserX Story* include personas, goals, interactions, contexts, and feedback. Nielsen's heuristics serve as the acceptance criteria. Expected user experience aspects can be specified as a heuristic. Based on these heuristics, the user experience could be estimated by extending and using a suitable method.

Joshi at al. [39] provide a *Usability Goals Achievement Metric (UGAM)*. This metric is calculated using individual parameters per usability quality (such as learnability, speed of use, and ease of use) weighted to a goal parameter score. This is the goal to be achieved.

After each usability evaluation, UX professionals calculate the achieved score based on the values from the usability evaluation. This makes it possible to determine whether the goal has been achieved by comparing the goal with the archived value. If the goal has not been achieved, it is possible to determine where it has not been achieved for each usability quality.

The two approaches are not directly based on product backlog items or requirements. Neither approach provides the possibility to estimate the user experience. In the end, both approaches can be used with an appropriate estimation method. Instead of the goal value, an estimated value of the user experience can be specified. The necessary prerequisites for a user experience value to be compared before development are given in both the approaches.

4.5. (RQ4) What retrospective proposals exist to improve the efficiency and effectiveness of the user experience process in terms of Agile software development?

To answer the fourth research question, we examined the included studies for approaches that improve the user experience process in its entirety, considering Agile software development. Thus, we identified two (6%) of the 33 studies that have been presented in the next two sections.

With the *Usability Goals Achievement Metric (UGAM)* developed by Joshi et al. [39], a goal for the user experience can be defined based on parameters or user experience aspects. By measuring after development, it can be determined to what extent the goal differs from actual reality. Further, the Metric *Index of Integration (IoI)* can be used to determine the maturity level of typical HCI activities in the development team. Both metrics (UGAM and IoI) can be used to perform a retrospective to identify the potential for improvement of the development of better user experience.

Nebe and Baloni [43] developed an Agile–HCD-Conformance Checklist based on the Checklist for User-Centeredness of Agile Processes of DIN ISO 9241-210 [2]. Besides, best practices from Agile human-centred design approaches were included in the checklist. This checklist can serve as a source for the evaluation of the team's user-centeredness of Agile processes. In addition to recording the evaluation, the list also provides recommendations for improvement based on findings from the integration of UCD and Agile processes.

5. Discussion

This SLR allows us to identify and classify approaches to the applicability of UX management. In Section 4, we have provided answers to our research questions. In this section, we discuss the results, which have been divided as the following: a discussion of the results in general and specifically for each research question in Section 5.1. And Section 5.2 discusses the limitation of the results based on the method used.

5.1. Meaning of findings

In this SLR, we have examined 49 studies. As already mentioned in Section 3, we did not limit the search to UX management, but extended it to UX and Agile methods in general. As a result, the search result is very widespread. The individual studies mainly describe the integration of UX methods into Agile software development in partially different ways. All the studies have the common goal to develop a better product or service with a high UX through the integration. We have used this goal as the basis for the answers to the research questions. None of the studies addressed UX management directly.

The ratio of the number of SLRs investigated in the remaining studies is rather interesting. A total of 16 SLRs (32%) and 33 (68%) other studies were investigated. The integration of UX methods in Agile software development has been addressed and discussed in the literature. However, there is a noticeable lack of sufficiently validated approaches that support integration. In the selected studies, we were able to investigate approaches that support integration but do not explicitly address UX management. We discuss this in detail in the next sections.

5.1.1. Findings related to RQ1

The *'Upfront UCD Design'* was identified as the most frequently used approach. This approach is criticized in some cases [55,60,72] because it does not resolve the matter of integration of UX methods directly in Agile software development. It is only an approach that coordinates the work of two teams — the UX team and the development team. In this respect, integration can only be described as limited.

All studies have more or less one thing in common: they all support the cooperation and communication between the team members. This is either a direct component and goal of the approach [55,60,70,72] or is achieved indirectly through the methods used. It should be noted that the team members usually consist of two rather different groups — the UX team and the development team. Both use different methods to do their work. For example, the UX team uses the methods listed in Table 5.

However, it turns out that using a single UX method is not the solution. Rather, it shows that UX methods are always integrated into a superordinate structure (framework, process, lifecycle, etc.). The superordinate structure provides an expected result, whereas the UX methods support the result. So, both are necessary.

5.1.2. Findings related to RQ2

As in the discussion of the first research questions, the answer to the second research question shows that the cooperation between the two disciplines is the primary approach. However, the most essential two findings can be summarized as follows:

1. UX methods or HCI and Agile methods have been developed independent of each other. There are similarities, such as iterations and feedback, but they were not developed with the intention that they could be integrated. Nevertheless, both methods are not mutually exclusive.
2. In management, UX is not perceived as a business strategy. This inevitably means that decisions regarding the use of UX methods within Agile software development must be made at the team level. In terms of the Agile method, this is feasible, but it will never have the value of a company-wide business strategy.

Therefore, future research studies should focus more on the integration of UX and Agile as well as on management in equal measure.

5.1.3. Findings related to RQ3

Two methods were found to answer the third research questions. These methods, *UserX Story* [42] and *Usability Goals Achievement Metric (UGAM)* [39], were not developed for estimating the user experience for a product backlog item, but they are basically suitable. However, further development of both methods is necessary to enable them to represent the estimated UX.

In essence, no approaches were found to estimate the user experience for a product backlog item or requirement before development. However, UX management needs a way to estimate or capture the UX before development (Section 2.2). Only then can it be measured after development whether the desire outcome has been achieved (Section 2.2).

5.1.4. Findings related to RQ4

The answers to the fourth and last research questions resulted in two approaches: *Usability Goals Achievement Metric (UGAM)* [39] in combination with *Index of Integration (IoI)* [39] and the *Agile–HCD Conformance Checklist* [43]. However, both approaches aim to determine the current status of integration and the application of UX methods. Although this is suitable for determining weaknesses and thus pointing out possible improvement potential, this is not explicitly addressed.

On the other hand, in Scrum, as the primary representative of Agile methods, the Sprint Retrospective [5] is a primary component. This team event should allow the team to improve each member or the team. All this will help to develop it more efficiently and in a better way in the next iteration. In none of the studies investigated was the Sprint Retrospective used to improve the UX processes or the UX methods used.

5.2. Limitation of the review

We have used a predefined protocol for conducting this study to ensure its completeness. We may not have identified some relevant studies due to the large number of existing studies. We have minimized this risk by using forward and backward snowballing. Owing to the lack of digital recording and organization of the studies in some cases, a residual risk cannot be ruled out.

Another possible weakness of our approach might be the chosen search string. We searched for UX methods in combination with Agile software development. However, it is quite conceivable that some existing UX have not been published in combination with Agile software development, even though they are suitable for the stated purpose.

Regarding the limitations of data extraction, we are aware that some aspects may not have been sufficiently documented in the studies analysed. For this reason, our results might have been different if the studies had been documented more accurately. We have tried to address this problem by conducting a comprehensive quality assessment of the studies included.

6. Conclusions and future work

This paper presents an SLR about managing the UX process to identify suitable approaches for user experience management. The SLR was conducted according to the guideline offered by Kitchenham and Charters [36]. In an initial search, we found 44,637 studies. Our search process reduced the number of studies to 1253. We analysed these studies by their titles and abstracts and performed a quality assessment. These measures helped us to select 49 studies, including seven studies that were selected through a snowballing process.

This SLR has different implications for both practitioners and scientists. Based on the explanations in ‘Related Work’ (Section 2), we can summarize that approaches and methods are used to develop a better user experience, but not goal-oriented by defining a UX goal. Furthermore, there is no definition of UX management or a common understanding of UX management.

The approaches identified in the studies deal with the integration of UX methods or HCI in Agile methods. Upfront UCD design, communication/collaboration, and teaching UX methods have been applied in several studies. All other approaches have only been presented in the respective study without being found in other studies. In addition to the use of the approaches, several UX methods have been identified that have been used in combination with the approaches. The three most frequently identified UX methods are prototyping (low/high), personas, and task/usage scenarios. Among the approaches and UX methods analysed, it has been found that only one approach makes it possible to define a UX goal before development and to test it after development [39]. All other approaches aimed to systematically integrate the UX methods into Agile software development. Many of them can be used for UX management if they are adapted accordingly.

In conclusion, it can be summarized that UX management is neither sufficiently described nor used in a targeted manner. Further research may focus on a process model or lifecycle that would take account of a desired UX goal and figure out whether the goal is reached. To this end, we will develop a UX lifecycle that includes a UX goal, a strategy to reach the UX goal, methods to interpret the outcome, and a UX retrospective to improve the usage of the UX lifecycle.

CRediT authorship contribution statement

Andreas Hinderks: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization, Project administration. **Francisco José Domínguez Mayo:** Methodology, Writing – review & editing, Supervision. **Jörg Thomaschewski:** Conceptualization, Validation, Writing – review & editing, Supervision. **María José Escalona:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was partially supported by the POLOLAS project (TIN2016-76956-C3-2-R) of the Spanish Government's Ministry of Economy and Competitiveness and by the NDT4.0 project (US-1251532) of the Junta de Andalucía, Spain.

References

[1] J. Preece, Y. Rogers, H. Sharp, Interaction Design: Beyond Human-Computer Interaction, fourth ed., Wiley, Chichester, 2015.

[2] ISO9241-210, Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems, 2020, ISO 9241-210:2020.

[3] M. Hassenzahl, The effect of perceived hedonic quality on product appealingness, Int. J. Hum.-Comput. Interact. 2001 (13(4)) (2001) 481–499.

[4] N. Tractinsky, Aesthetics and apparent usability, in: S. Pemberton (Ed.), The SIGCHI Conference, 1997, pp. 115–122, http://dx.doi.org/10.1145/258549.258626.

[5] K. Schwaber, Agile Project Management with Scrum, in: Microsoft professional, Microsoft Press, Redmond, Wash, 2004.

[6] D.J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, Sequim, Washington, 2010.

[7] K. Beck, C. Andres, Extreme Programming Explained: Embrace Change, second ed., 6. printing, in: The XP series, Addison-Wesley, Boston, 2007.

[8] P. Serrador, J.K. Pinto, Does Agile work? — A quantitative analysis of agile project success, Int. J. Project Manag. 33 (5) (2015) 1040–1051, http://dx.doi.org/10.1016/j.ijproman.2015.01.006.

[9] B. Boehm, R. Turner, Using risk to balance agile and plan- driven methods, Computer 36 (6) (2003) 57–66, http://dx.doi.org/10.1109/MC.2003.1204376.

[10] P.F. Drucker, The Practice of Management, HarperCollins, New York, NY, 2009.

[11] J. Magretta, N.D. Stone, What Management Is: How It Works and Why It's Everyone's Business, Profile Books, London, 2013.

[12] M. Mckeown, The Strategy Book: How to Think and Act Strategically to Deliver Outstanding Results, third ed., Pearson, Harlow, England, 2020.

[13] M. Schrepp, J. Thomaschewski, Handbook for the Modular Extension of the User Experience Questionnaire, 2019.

[14] J. Sauro, SUPR-Q: A comprehensive measure of the quality of the website user experience, J. Usability Stud. 2015 (10) (2015) 68–86.

[15] A. Szóstek, A look into some practices behind microsoft UX management, in: J.A. Konstan, E.H. Chi, K. Höök (Eds.), Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts - CHI EA '12, ACM Press, New York, New York, USA, 2012, p. 605, http://dx.doi.org/10.1145/2212776.2212833.

[16] R.I. Anderson, J. Ashley, T. Herrmann, J. Miller, J. Nieters, S.S. Eves, S.T. Watson, Moving ux into a position of corporate influence, in: M.B. Rosson, D. Gilmore (Eds.), CHI '07 Extended Abstracts on Human Factors in Computing Systems - CHI '07, ACM Press, New York, New York, USA, 2007, p. 1905, http://dx.doi.org/10.1145/1240866.1240920.

[17] D. Rosenberg, The business of UX management, Interactions 26 (3) (2019) 28–35, http://dx.doi.org/10.1145/3318131.

[18] H.B.-L. Duh, J.-J. Lee, P.L.P. Rau, M.Q. Chen, The management model development of user experience design in organization, in: P.-L.P. Rau (Ed.), Cross-Cultural Design, in: Lecture Notes in Computer Science, vol. 9741, Springer International Publishing, Cham, 2016, pp. 163–172, http://dx.doi.org/10.1007/978-3-319-40093-8_17.

[19] D. Salah, R. Paige, P. Cairns, Integrating agile development processes and user centred design- A place for usability maturity models? in: S. Sauer, C. Bogdan, P. Forbrig, R. Bernhaupt, M. Winckler (Eds.), Human-Centered Software Engineering, in: Lecture Notes in Computer Science, vol. 8742, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 108–125, http://dx.doi.org/10.1007/978-3-662-44811-3_7.

[20] A. Bruun, Training software developers in usability engineering, in: E. Hvannberg, M.K. Lárusdóttir, A. Blandford, J. Gulliksen (Eds.), Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries - NordiCHI '10, ACM Press, New York, New York, USA, 2010, p. 82, http://dx.doi.org/10.1145/1868914.1868928.

[21] T. Silva da Silva, A. Martin, F. Maurer, M. Silveira, User-centered design and agile methods: A systematic review, in: 2011 AGILE Conference, IEEE, 2011, pp. 77–86, http://dx.doi.org/10.1109/AGILE.2011.24.

[22] D. Salah, R.F. Paige, P. Cairns, A systematic literature review for agile development processes and user centred design integration, in: M. Shepperd, T. Hall, I. Myrtveit (Eds.), Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14, ACM Press, New York, New York, USA, 2014, pp. 1–10, http://dx.doi.org/10.1145/2601248.2601276.

[23] 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, in: 2014 Agile Conference, IEEE, 2014, pp. 24–32, http://dx.doi.org/10.1109/AGILE.2014.17.

[24] C. Salvador, A. Nakasone, J.A. Pow-Sang, A systematic review of usability techniques in agile methodologies, in: Unknown (Ed.), Proceedings of the 7th Euro American Conference on Telematics and Information Systems, in: EATIS, vol. 14, ACM Press, New York, New York, USA, 2014, pp. 1–6, http://dx.doi.org/10.1145/2590651.2590668.

[25] T. Silva 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.), Computational Science and Its Applications, ICCSA 2015, in: Lecture Notes in Computer Science, vol. 9159, Springer International Publishing, Cham, 2015, pp. 86–100, http://dx.doi.org/10.1007/978-3-319-21413-9_7.

[26] 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, http://dx.doi.org/10.1016/j.infsof.2015.01.004.

[27] D.A. Magües, J.W. Castro, S.T. Acuña, Usability in agile development: A systematic mapping study, in: 2016 XLII Latin American Computing Conference, CLEI, IEEE, 2016, pp. 1–8, http://dx.doi.org/10.1109/CLEI.2016.7833347.

[28] L. Caballero, A.M. Moreno, A. Seffah, How agile developers integrate user-centered design into their processes: A literature review, Int. J. Softw. Eng. Knowl. Eng. 26 (08) (2016) 1175–1201, http://dx.doi.org/10.1142/S0218194016500418.

[29] D.A. Magues, J.W. Castro, S.T. Acuna, HCI usability techniques in agile development, in: 2016 IEEE International Conference on Automatica, ICA-ACCA, IEEE, 2016, pp. 1–7, http://dx.doi.org/10.1109/ICA-ACCA.2016.7778513.

[30] E.-M. Schön, J. Thomaschewski, M.J. Escalona, Agile requirements engineering: A systematic literature review, Comput. Stand. Interfaces 49 (2017) 79–91, http://dx.doi.org/10.1016/j.csi.2016.08.011.

[31] A. Garcia, T. Silva da Silva, M. Selbach Silveira, Artifacts for agile user-centered design: A systematic mapping, in: Proceedings of the 50th Hawaii International Conference on System Sciences (2017), in: Proceedings of the Annual Hawaii International Conference on System Sciences, Hawaii International Conference on System Sciences, 2017, http://dx.doi.org/10.24251/HICSS.2017.706.

[32] R. Hoda, N. Salleh, J. Grundy, H.M. Tee, Systematic literature reviews in agile software development: A tertiary study, Inf. Softw. Technol. 85 (2017) 60–70, http://dx.doi.org/10.1016/j.infsof.2017.01.007.

[33] J.C. Pereira, R.d.F. Russo, Design thinking integrated in agile software development: A systematic literature review, Procedia Comput. Sci. 138 (2018) 775–782, http://dx.doi.org/10.1016/j.procs.2018.10.101.

[34] T.S. Da Silva, M.S. Silveira, F. Maurer, F.F. Silveira, The evolution of agile UXD, Inf. Softw. Technol. 102 (2018) 1–5, http://dx.doi.org/10.1016/j.infsof.2018.04.008.

[35] K. Curcio, R. Santana, S. Reinehr, A. Malucelli, Usability in agile software development: A tertiary study, Comput. Stand. Interfaces 64 (2019) 61–77, http://dx.doi.org/10.1016/j.csi.2018.12.003.

[36] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, 2007.

[37] A. Hinderks, M. Schrepp, F.J. Domínguez Mayo, M.J. Escalona, J. Thomaschewski, UEQ KPI Value Range based on the UEQ Benchmark, http://dx.doi.org/10.13140/RG.2.2.34239.76967.

[38] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: M. Shepperd, T. Hall, I. Myrtveit (Eds.), Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14, ACM Press, New York, New York, USA, 2014, pp. 1–10, http://dx.doi.org/10.1145/2601248.2601268.

[39] A. Joshi, N.L. Sarda, S. Tripathi, Measuring effectiveness of HCI integration in software development processes, J. Syst. Softw. 83 (11) (2010) 2045–2058, http://dx.doi.org/10.1016/j.jss.2010.03.078.

[40] L. Caballero, A.M. Moreno, A. Seffah, Persona as a tool to involving human in agile methods: Contributions from HCI and marketing, in: S. Sauer, C. Bogdan, P. Forbrig, R. Bernhaupt, M. Winckler (Eds.), Human-Centered Software Engineering, in: Lecture Notes in Computer Science, vol. 8742, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 283–290, http://dx.doi.org/10.1007/978-3-662-44811-3_20.

[41] D. Distante, A. Garrido, J. Camelier-Carvajal, R. Giandini, G. Rossi, Business processes refactoring to improve usability in E-commerce applications, Electron. Commerce Res. 14 (4) (2014) 497–529, http://dx.doi.org/10.1007/s10660-014-9149-0.

[42] J. Choma, L.A.M. Zaina, D. Beraldo, UserX story: Incorporating UX aspects into user stories elaboration, in: M. Kurosu (Ed.), Human-Computer Interaction. Theory, Design, Development and Practice, in: Lecture Notes in Computer Science, vol. 9731, Springer International Publishing, Cham, 2016, pp. 131–140, http://dx.doi.org/10.1007/978-3-319-39510-4_13.

[43] K. Nebe, S. Baloni, Agile human-centred design: A conformance checklist, in: S. Yamamoto (Ed.), Human Interface and the Management of Information: Information, Design and Interaction, in: Lecture Notes in Computer Science, vol. 9734, Springer International Publishing, Cham, 2016, pp. 442–453, http://dx.doi.org/10.1007/978-3-319-40349-6_42.

[44] S. Firmenich, A. Garrido, J. Grigera, J.M. Rivero, G. Rossi, Usability improvement through A/B testing and refactoring, Softw. Qual. J. 27 (1) (2019) 203–240, http://dx.doi.org/10.1007/s11219-018-9413-y.

[45] T. Memmel, F. Gundelsweiler, H. Reiterer, Agile human-centered software engineering, in: Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...But Not As We Know It - Vol. 1, in: BCS-HCI '07, BCS Learning & Development Ltd, Swindon, GBR, 2007, pp. 167–175.

[46] P. Wolkerstorfer, M. Tscheligi, R. Sefelin, H. Milchrahm, Z. Hussain, M. Lechner, S. Shahzad, Probing an agile usability process, in: M. Czerwinski, A. Lund, D. Tan (Eds.), Proceeding of the Twenty-Sixth Annual CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI '08, ACM Press, New York, New York, USA, 2008, p. 2151, http://dx.doi.org/10.1145/1358628.1358648.

[47] M. Singh, U-SCRUM: An agile methodology for promoting usability, in: Agile 2008 Conference, IEEE, 2008, pp. 555–560, http://dx.doi.org/10.1109/Agile.2008.33.

[48] Y. Xiong, A. Wang, A new combined method for UCD and software development and case study, in: The 2nd International Conference on Information Science and Engineering, IEEE, 2010, pp. 1–4, http://dx.doi.org/10.1109/ICISE.2010.5690032.

[49] S.R. Humayoun, Y. Dubinsky, T. Catarci, A three-fold integration framework to incorporate user–centered design into agile software development, in: M. Kurosu (Ed.), Human Centered Design, in: Lecture Notes in Computer Science, vol. 6776, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 55–64, http://dx.doi.org/10.1007/978-3-642-21753-1_7.

[50] T. Silva da Silva, M. Selbach Silveira, F. Maurer, T. Hellmann, User experience design and agile development: From theory to practice, J. Softw. Eng. Appl. 05 (10) (2012) 743–751, http://dx.doi.org/10.4236/jsea.2012.510087.

[51] A.L. Peres, T.S.D. Silva, F.S. Silva, F.F. Soares, C.R.M.D. Carvalho, S.R.D.L. Meira, Agileux model: Towards a reference model on integrating UX in developing software using agile methodologies, in: 2014 Agile Conference, IEEE, 2014, pp. 61–63, http://dx.doi.org/10.1109/AGILE.2014.15.

[52] S.M. Butt, A. Onn, M.M. Butt, N.T. Inam, S.M. Butt, Incorporation of usability evaluation methods in agile software model, in: 17th IEEE International Multi Topic Conference 2014, IEEE, 2014, pp. 193–199, http://dx.doi.org/10.1109/INMIC.2014.7097336.

[53] P.L.M. Navarro, G.M. Pérez, D.S. Ruiz, A script-based prototyping framework to boost agile-UX developments, J. Comput. Sci. Tech. 31 (6) (2016) 1246–1261, http://dx.doi.org/10.1007/s11390-016-1695-6.

[54] M. Aguilar, C. Zapata, Integrating UCD and an agile methodology in the development of a mobile catalog of plants, in: M. Soares, C. Falcão, T.Z. Ahram (Eds.), Advances in Ergonomics Modeling, Usability & Special Populations, in: Advances in Intelligent Systems and Computing, vol. 486, Springer International Publishing, Cham, 2017, pp. 75–87, http://dx.doi.org/10.1007/978-3-319-41685-4_8.

[55] J. Ferreira, J. Noble, R. Biddle, Agile development iterations and UI design, in: AGILE 2007, AGILE 2007, IEEE, 2007, pp. 50–58, http://dx.doi.org/10.1109/AGILE.2007.8.

[56] T. Øvad, N. Bornoe, L.B. Larsen, J. Stage, Teaching software developers to perform UX tasks, in: B. Ploderer, M. Carter, M. Gibbs, W. Smith, F. Vetere (Eds.), Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction on - OzCHI '15, ACM Press, New York, New York, USA, 2015, pp. 397–406, http://dx.doi.org/10.1145/2838739.2838764.

[57] T. Øvad, L.B. Larsen, How to reduce the UX bottleneck – train your software developers, Behav. Inf. Technol. 35 (12) (2016) 1080–1090, http://dx.doi.org/10.1080/0144929X.2016.1225818.

[58] L. Schwartz, Agile-user experience design: an agile and user-centered process? in: ICSEA 2013: The Eighth International Conference on Software Engineering Advances, 2013, pp. 346–351.

[59] L. Schwartz, Agile-user experience design: Does the involvement of usability experts improve the software quality? in: International Journal on Advances in Software, Vol. 7, 2014.

[60] M. Larusdottir, J. Gulliksen, A. Cajander, A license to kill – improving UCSD in agile development, J. Syst. Softw. 123 (2017) 214–222, http://dx.doi.org/10.1016/j.jss.2016.01.024.

[61] D. Fox, J. Sillito, F. Maurer, Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry, in: Agile 2008 Conference, IEEE, 2008, pp. 63–72, http://dx.doi.org/10.1109/Agile.2008.78.

[62] K. Kuusinen, Improving UX work in scrum development: A three-year follow-up study in a company, in: S. Sauer, C. Bogdan, P. Forbrig, R. Bernhaupt, M. Winckler (Eds.), Human-Centered Software Engineering, in: Lecture Notes in Computer Science, vol. 8742, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 259–266, http://dx.doi.org/10.1007/978-3-662-44811-3_17.

[63] T.S.D. Silva, M.S. Silveira, F. Maurer, Usability evaluation practices within agile development, in: 2015 48th Hawaii International Conference on System Sciences, IEEE, 2015, pp. 5133–5142, http://dx.doi.org/10.1109/HICSS.2015.607.

[64] L. Hokkanen, K. Kuusinen, K. Väänänen, Minimum viable user experience: A framework for supporting product design in startups, in: H. Sharp, T. Hall (Eds.), Agile Processes, in Software Engineering, and Extreme Programming, in: Lecture Notes in Business Information Processing, vol. 251, Springer International Publishing, Cham, 2016, pp. 66–78, http://dx.doi.org/10.1007/978-3-319-33515-5_6.

[65] P. Kashfi, A. Nilsson, R. Feldt, Integrating user experience practices into software development processes: implications of the UX characteristics, PeerJ Comput. Sci. 3 (2017) e130, http://dx.doi.org/10.7717/peerj-cs.130.

[66] N. Pillay, J. Wing, Agile UX: Integrating good UX development practices in agile, in: 2019 Conference on Information Communications Technology and Society, ICTAS, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICTAS.2019.8703607.

[67] M.K. Larusdottir, Usability evaluation in software development practice, in: D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, M. Winckler (Eds.), Human-Computer Interaction – INTERACT 2011, in: Lecture Notes in Computer Science, vol. 6949, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 430–433, http://dx.doi.org/10.1007/978-3-642-23768-3_50.

[68] Y. Jia, M.K. Larusdottir, A. Cajander, The usage of usability techniques in scrum projects, in: D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, M. Winckler, P. Forbrig, R. Bernhaupt (Eds.), Human-Centered Software Engineering, in: Lecture Notes in Computer Science, vol. 7623, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 331–341, http://dx.doi.org/10.1007/978-3-642-34347-6_25.

[69] K. Kuusinen, K. Väänänen-Vainio-Mattila, How to make agile UX work more efficient, in: L. Malmborg, T. Pederson (Eds.), Proceedings of the 7th Nordic Conference on Human-Computer Interaction Making Sense Through Design - NordiCHI '12, ACM Press, New York, New York, USA, 2012, p. 139, http://dx.doi.org/10.1145/2399016.2399037.

[70] J. Ferreira, H. Sharp, H. Robinson, Agile development and user experience design integration as an ongoing achievement in practice, in: 2012 Agile Conference, IEEE, 2012, pp. 11–20, http://dx.doi.org/10.1109/Agile.2012.33.

[71] T. Øvad, L.B. Larsen, Templates: A key to success when training developers to perform UX tasks, in: G. Cockton, M. Lárusdóttir, P. Gregory, A.s. Cajander (Eds.), Integrating User-Centred Design in Agile Development, in: Human–Computer Interaction Series, Springer International Publishing, Cham, 2016, pp. 77–96, http://dx.doi.org/10.1007/978-3-319-32165-3_3.

[72] K. Kuusinen, Task allocation between UX specialists and developers in agile software development projects, in: J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, M. Winckler (Eds.), Human-Computer Interaction – INTERACT 2015, in: Lecture Notes in Computer Science, vol. 9298, Springer International Publishing, Cham, 2015, pp. 27–44, http://dx.doi.org/10.1007/978-3-319-22698-9_3.