Challenges of Integrating Agile and UX/UCD: Systematic Literature Review

Muna Al-Razgan
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

Razan Aldossari
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

Luluah Albesher
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

Maha Alshammari
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

Shahad Alotaibi
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

Nourah Alanquary
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

Norah Alshahrani
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

Lama Alsaykhan
Software Engineering Department
King Saud University
Riyadh 11345, Saudi Arabia

ABSTRACT

The integration of Agile development with User Experience (UX) and with User-Centered Design (UCD) grabs IT companies' attention and has become one of the most significant topics in the last few years. In agile UX/UCD processes, the software development process is positively impacted. This systematic review of the challenges of integrating Agile with UX/UCD is provided here to assist other researchers in identifying the recent major challenges associated with Agile and UX/UCD. In addition, this review covers articles available in five research databases. Based on the search strategy, 70 studies were identified relevant to the scope from 2001 to 2021. The findings of this review provide insight into the research area of Agile UX/UCD and offer future research directions.

Keywords

agile software development, user-centered design, systematic review, agile user center design integration, agile user center design challenges

1. INTRODUCTION

As part of the software development field, Agile methods have become increasingly popular since the 1990s. This is mainly due to the failures of other approaches, such as Waterfall and Parallel methodologies, which failed to address the complexity and size of projects [1]. In addition, Agile methods have two significant advantages: speed and communication [2]. Over the past two decades, Agile and UX have always been closely associated. Each of these disciplines is interdependent or interrelated with the other, and both are very useful in the field of software engineering through building efficient and high-quality software [3]. However, the integration of Agile and UX teams is one of the most successful and significant trends in the field. At the same time, an increasing understanding of the importance of good UX evolved, and the need to integrate these two areas emerged.

Agile methods primarily refer to activities relating to code creation or project management, and the Agile Manifesto declares: "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software" [4]. The UX design methods, on the other hand, describe activities that focus on the product's interactions and interface with users. Moreover, Agile focuses on the development of usable software while ignoring usability and the involvement of the end user during the development phase, whereas usability approaches focus on the user's needs to develop usable software [5].

This systematic literature review aims to identify the challenges associated with integrating the two concepts and explore how those challenges have been addressed. In addition, the review describes how recent studies have combined these two concepts. In particular, this study aims to explore the most used UCD techniques in agile development. In addition, it attempts to examine the agile methods that adopt UCD techniques. The remaining sections of the paper are structured as follows: Section II describes the methods used to conduct this study. Within the same section introduces the research objectives, research questions, and review method, which includes a description of the search strategy, selection process, and quality assessment. Section III discusses how the articles will be classified. Section IV summarizes the main findings of the study. Hence, it summarizes the included studies and answers to the research questions. Finally, section V discusses the significance of the findings and the study's future works and limitations.

2. METHODOLOGY

This section presents the method used in this study. First, research questions are provided. Then the research strategy is explained as well as the inclusion criteria.

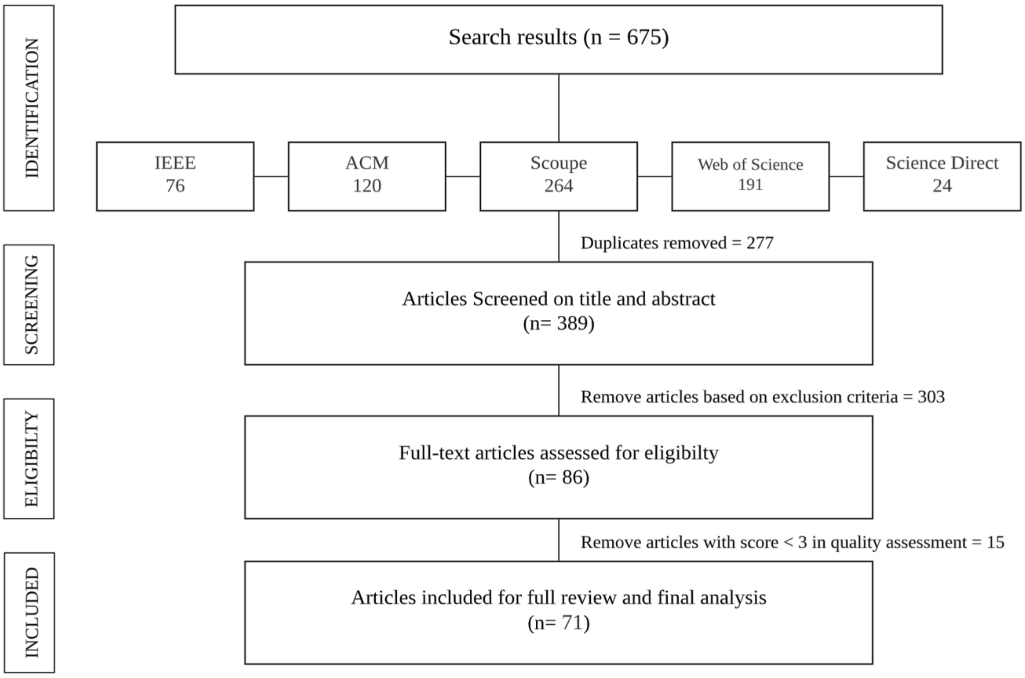

IDENTIFICATION
Search results (n = 675)
IEEE 76
ACM 120
Scoupe 264
Web of Science 191
Science Direct 24
SCREENING
Duplicates removed = 277
Articles Screened on title and abstract (n= 389)
Remove articles based on exclusion criteria = 303
ELIGIBILTY
Full-text articles assessed for eligibilty (n= 86)
Remove articles with score < 3 in quality assessment = 15
INCLUDED
Articles included for full review and final analysis (n= 71)

Figure 1. PRISMA flow chart.

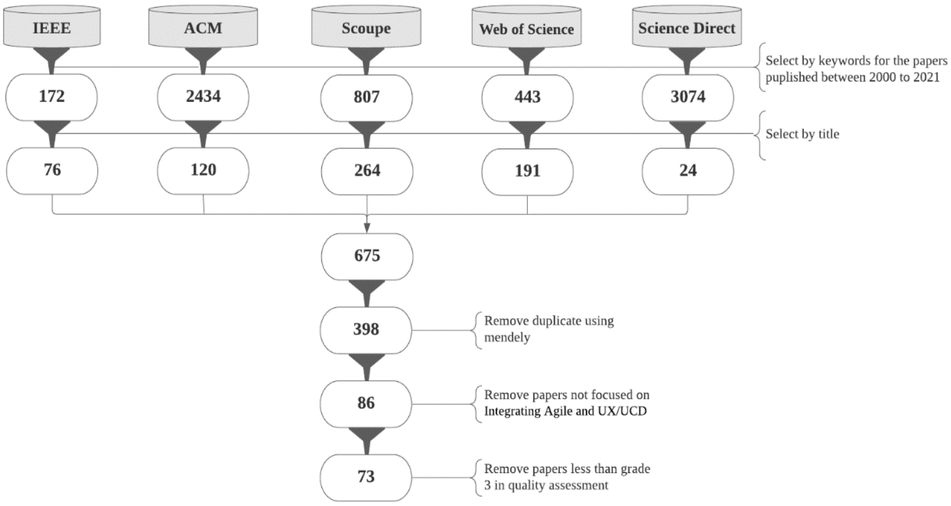

IEEE
ACM
Scoupe
Web of Science
Science Direct
172
2434
807
443
3074
Select by keywords for the papers puplished between 2000 to 2021
76
120
264
191
24
Select by title
675
398
Remove duplicate using mendely
86
Remove papers not focused on Integrating Agile and UX/UCD
73
Remove papers less than grade 3 in quality assessment

Figure 2. Search and selection process.

This section then concludes with the quality assessment process. In addition, the systematic review of literature was performed by applying the preferred reporting items for systematic reviews and meta-analyses (PRISMA) protocol [6]. Figure 1 provides details of the PRISMA flow chart, whereas Figure 2 demonstrates the detailed search and selection process. The charts presented in this section were created through Lucidchart [7].

2.1 Research questions

To conduct this systematic review, the first step was to identify the research questions. This study was undertaken to review recent studies in the area of integrating agile and UX/UCD with a focus on the challenges. Therefore, four research questions were defined as follows:

- RQ1: What are the most used UCD techniques in agile development?
- RQ2: What are the agile methods that adopt UCD techniques?
- RQ3: How does the integration among the agile method and UX/UCD improve the process?
- RQ4: What are the challenges of the integration of Agile and UX/UCD?

RQ1 aims to gain insight into different UCD/UX techniques that have been applied in agile. RQ2 aims to gain insight into different agile approaches that have been combined with UCD practices. RQ3 aims to investigate how processes can be improved when Agile and UX/UCD are integrated. RQ4 aims to identify challenges associated with integrating Agile and UX/UCD.

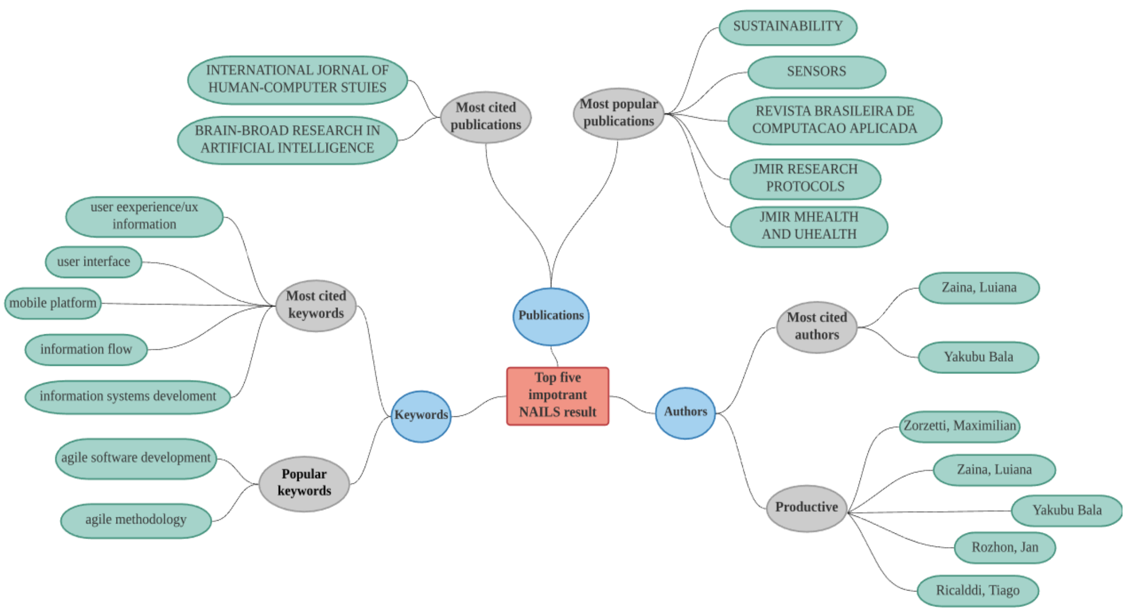


Figure 3. Summary of NAILS results for Agile and UX/UCD publications for 200-2021

2.2 Search Strategy

The network analysis interface for literature studies (NAILS) [8] was utilized to gain an overview of the publications in the Agile and UX/UCD field. NAILS is a free open-source software used to analyze literature studies. It has been used to analyze about 454 articles published on agile and UX/UCD from 2000 to 2021, as illustrated in Figure 3. Moreover, NAILS software also analyses publication information from the Web of Science and delivers detailed information on publication timeline. The literature on integrating Agile and UX/UCD expanded in 2013. The software also provides detailed information on the authors in the field, such as the most prominent and most-cited authors in the field. Such information can be sorted by the total number of citations and published articles. It also shows the important articles according to the most popular and most cited articles. In addition, the software provides essential keywords, i.e., the most popular and most cited. These keywords are normally sorted by the number of articles in which they were mentioned as well as by their total number of citations. Furthermore, the sorting of articles can be performed by the top 25 through three measures of significance: i) in-degree in the citation network, ii) citation count obtained from the Web of Science, and iii) PageRank score in the citation network.

Following the overview analysis, the search was conducted by searching for conferences and journal articles from five databases, which included:

- IEEE Xplore Digital Library
- ACM Digital Library
- Scopus
- The Web of Science
- Science Direct

The specific search in each database was by using titles with keywords as in the following: ("Agile software development" OR "Agile methodology" OR "Scrum" OR "Extreme Programming" OR "Agile") AND ("user experience" OR "user interface" OR "user-centered design").

2.3 Study selection and inclusion criteria

The results of this study were obtained and sorted using

Mendeley Reference Management Software. The inclusion and exclusion criteria were identified for the articles in the present study as illustrated in Figure 4. The inclusion criteria were:

- The article was relevant to Challenges of Integrating Agile (UX/UCD).
- The language examined in the Agile article was English.
- The article focus was related to the RQs.
- Article published during the period 2000–2021.
- Studies available in full text and retrievable online.
- Articles with a quality assessment grade of at least three (As defined in this study).

These criteria were applied to filter the articles. The research focused on recent articles in integration of Agile and UX/UCD; therefore, the first criterion was to include articles published during the period 2000–2021. This criterion was applied using the databases' research boundaries. The second criterion was to filter the articles that demonstrate the integration of UCD/UX processes with Agile methodologies. In the articles that examine the challenges in the integrating agile and UCD/UX, the article's focus had to be related to the RQs. This criterion was utilized using Mendeley software. For integration articles that investigated challenges, the focus of the article was on the RQs. In regards to this criterion, Rayyan [9] was utilized. Rayyan is a web application that presents the published article and the abstract that helped the authors of this systematic review. This is performed to collaborate and vote on articles according to the RQ criteria. There were three voting options: include, exclude, and maybe. In addition, Rayyan allows hiding the voting of individuals from other users. Seven authors utilized the Rayyan website for the purpose of evaluation, and each article was voted on anonymously by seven users. Each criterion was performed separately as illustrated in Figure 2. All articles that fit the other criteria and which received three "include" votes and "exclude" and one 'maybe" were ruled out. Articles that had five "maybe" votes and one "include" and one "exclude" were further checked by a first reviewer. In these cases, the first reviewer makes the final decision regarding the inclusion or exclusion of the article.

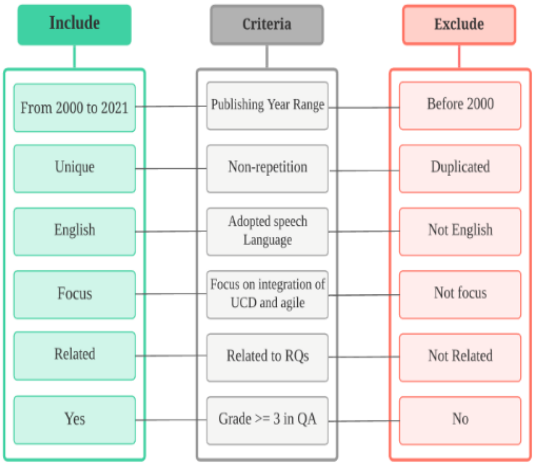


Figure 4. Search and selection process.

2.4 Quality Assessment

The quality assessment process shown in Fig. 5 was based on the following predefined quality questions

- Are the aims of the research clearly stated?
- Does the paper provide integration of Agile and (UX/USD)?
- Does the Agile integration with (UX/USD) improve effort?
- Does the article provide answers to the formulated RQs?

Each quality assessment question answered in the affirmative was given one point against the quality score. All authors were included and evaluated the articles based on the quality questions and related research questions. Research articles with a quality score greater than or equal to three were selected. The assessment process was as follows: If the article addressed the question completely, it was given one point, and 0.5 points if the question was answered partially, whereas zero point was given to articles that did not answer the question. Since the fourth question was a composite, it was further divided into sub-questions according to the main research questions. Each sub-question had a score, which was then divided by four to obtain one overall score. Following that, the points were summed for all quality questions. If the research article obtained a non-integer total score, it was then rounded to the nearest digit. For example, a 3.4 would be rounded to 3. Only research articles with total scores of three and above were considered in the criterion.

3. CHARACTERISTICS OF THE ARTICLES

To cover the trends of the articles, Figure 6. Below illustrate the selected articles for the current study according to the year of publication. The characteristics of the reviewed articles are provided in Table 1.

3.1 Classification of Articles Based on Techniques

1) **Little Design Up Front:** Several design techniques have been adopted into Agile development, such as Little Design up Front as the commonly used practices for upfront design was the “sprint zero” approach. In [10], the author used persona at the beginning of the project, particularly in cycle zero. According to the author, this assists with summarizing and describing target users and workflows in brief and vivid terms. In [11], the authors described how a one-day design studio fuses agile development team practice with user-centered design. They conducted a case study to examine the concept of a studio approach to interaction design in the context of agile User Experience Design. By using a design studio approach, the entire team was able to gain a solid understanding of the design so that they could start development and to strengthen their collaboration. A pre-development usability evaluation was conducted in [12] as during Sprint 0, user stories and high-fidelity prototypes were created in addition to identifying the key users. According to [13] ,the Enhanced Agile Process (EAP) study followed the same model as the first Current Agile Process (CAP) study, with a user experience designer working as part of the agile team to design and analyze LDUF objects each iteration. The UX designer collaborated closely with the agile team and the project manager throughout the software development process. Product owners must develop and evaluate the design artifacts to assist in the analysis, verification, and validation. By the end of the third iteration, the product team formally assessed the agile team's end product and signed it off.

2) Prototypes: Prototypes were used for a variety of purposes, but mainly to test a design. In [14], the authors used low-fidelity paper and high-fidelity prototypes to perform evaluations and usability tests. Similarly, another study [15] developed a technique that combines prototyping with two essential software quality criteria, usability and user experience (UX). As a result, this technique was created to be used by agile teams regardless of the number of resources available, the size of the project, or the agile methodology used. The researchers used a prototyping technique in [16] to make it easier for the team to understand the design when seeing the prototype sketch. Therefore, prototypes are great tools for making design decisions when multiple design solutions are available. For this reason, they conducted two types of tests in the initial phase, prototype testing and iterative redesign usability evaluation, involving eight research team members. The results showed that prototyping and usability heuristics are presented as a fast and cost-efficient yet still effective and accurate method of evaluating the user experience of educational software.

A user-interface test-driven development approach is proposed in [17], whereby tests are recorded and replayed from prototypes using a capture-replay tool. As low-fidelity prototypes can be created quickly and at a low cost, they are perfect for iterative agile evaluations. Two benefits can be gained from this approach: First, usability concerns will be discovered early in the development process. Second, if the

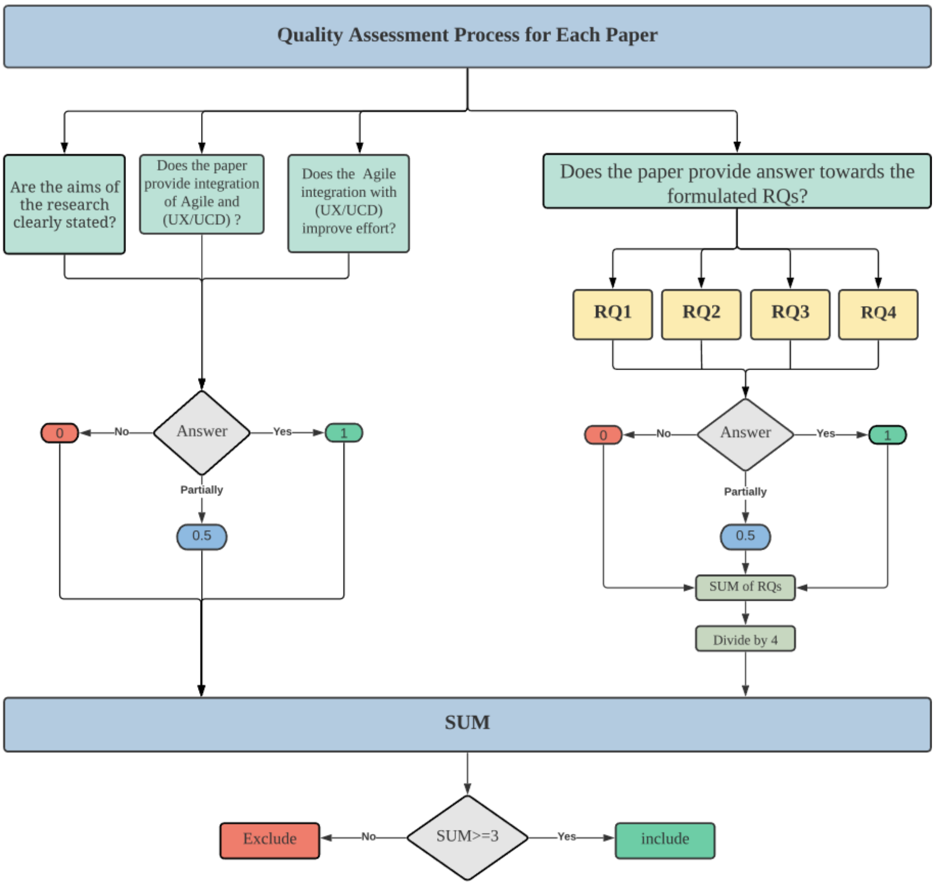

Quality Assessment Process for Each Paper
Are the aims of the research clearly stated?
Does the paper provide integration of Agile and (UX/UCD) ?
Does the Agile integration with (UX/UCD) improve effort?
Does the paper provide answer towards the formulated RQs?
RQ1
RQ2
RQ3
RQ4
Answer
No
Yes
0
1
Partially
0.5
Answer
No
Yes
0
1
Partially
0.5
SUM of RQs
Divide by 4
SUM
SUM>=3
No
Yes
Exclude
include

Figure 5. Quality assessment process.

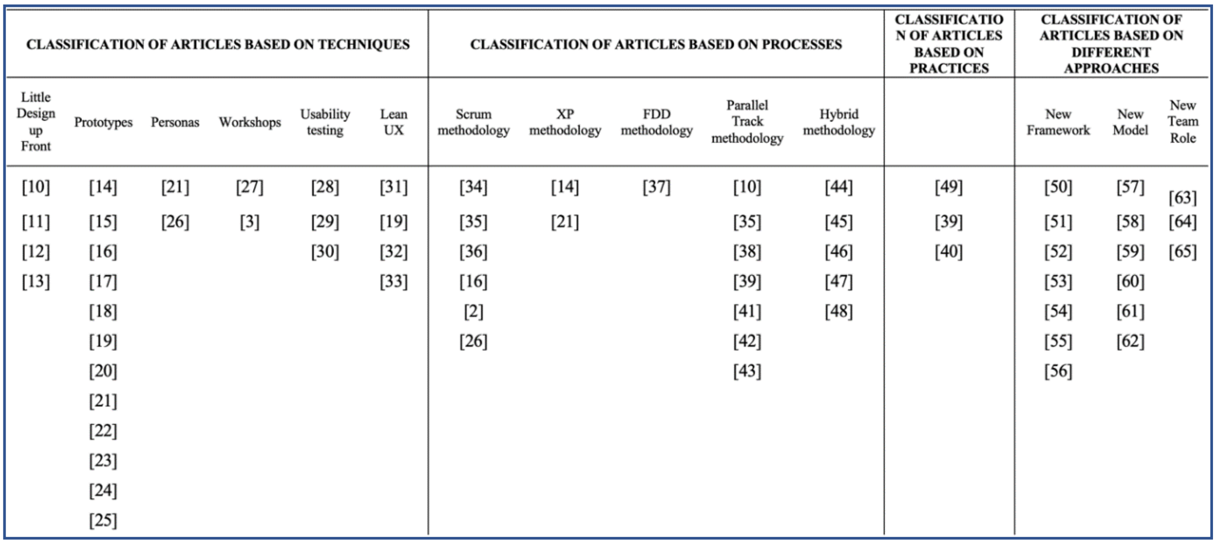
prototype is decorated with automated information that identifies widgets and makes assertions about them, tests can be recorded and replayed into the actual GUI during implementation. With this approach, it is easier to write GUI tests without needing a GUI first. It is possible to use the approach for User Interface Test-Driven Development. In contrast, no conclusions can yet be drawn regarding its practicality or usefulness.

Researchers used prototypes in [18] to perform usability evaluations, usability inspections, and user testing. Similarly, the usability evaluation and the user-centered design process are iterative processes, and agile methods support these activities by improving design and prioritizing tasks. The purpose of the suggested methodology is to conduct more usability inspection evaluations rather than repeating user testing sessions too often. As a result, product development can be driven and is more effective than having usability evaluation results at the end of development. This will ultimately result in a higher quality product with lower costs.

In [19], the researchers used prototypes to investigate an interactive interface's interaction design and usability to solve addition and subtraction problems for children with learning disabilities: Down syndrome (DS). Prototypes facilitate creating applications that focus on users, enabling to obtain continuous feedback. The primary goal of the suggested methodology is to identify usability issues and user experience during the early phases of educational software development. The researchers claim that applying agile human-centered methodologies to this particular case solved several issues concerning flexibility and efficiency, design, support, languages and sounds, consistency, error prevention, and other factors that may affect the satisfaction of specific users. In [20],the author presented an early-stage prototype focused on usability principles for interactive systems in an Ethiopian software development company, especially in poor villages. For this segment of the populace to utilize ICT services, utility and usability are critical as a user interface feature must have specific characteristics such as efficiency, learnability, memorability, error-free operation, and a pleasant appearance. Prototypes and quick delivery of operational versions assisted in gathering user feedback, and user involvement was beneficial for interactive systems.

A Low-fidelity prototype was utilized in [21] which nine screens were identified and designed as part of the screen interface. In order for this prototype to become a real system, it must be usable by users who can provide feedback. Therefore, the prototype must reflect actual simulated user scenarios. Consequently, the application was successfully implemented according to the project scope and methodology. In addition, usability has grown virtually as the users can use the application without assistance or instructions. In [22], the authors attempted to design a mobile prototype to assist architects during their visits to construction sites to define user requirements from the new prototype. To complement their strategy, they enrolled the assistance of business professionals and implemented Agile UX. In [23], the product owner team and the UX team collaborated on developing a mobile app with usability features, a task for the product owner to meet with the UX team face to face to clarify the basic needs as a result of the meeting. Understanding high-level user-focused project goals. A high-fidelity prototype was generally used to illustrate the user interface. Hence, business

Table 1. Classification-wise breakdown of the reviewed articles



CLASSIFICATION OF ARTICLES BASED ON TECHNIQUES						CLASSIFICATION OF ARTICLES BASED ON PROCESSES					CLASSIFICATION OF ARTICLES BASED ON PRACTICES	CLASSIFICATION OF ARTICLES BASED ON DIFFERENT APPROACHES		
Little Design up Front	Prototypes	Personas	Workshops	Usability testing	Lean UX	Scrum methodology	XP methodology	FDD methodology	Parallel Track methodology	Hybrid methodology		New Framework	New Model	New Team Role
[10]	[14]	[21]	[27]	[28]	[31]	[34]	[14]	[37]	[10]	[44]	[49]	[50]	[57]	[63]
[11]	[15]	[26]	[3]	[29]	[19]	[35]	[21]		[35]	[45]	[39]	[51]	[58]	[64]
[12]	[16]			[30]	[32]	[36]			[38]	[46]	[40]	[52]	[59]	[65]
[13]	[17]				[33]	[16]			[39]	[47]		[53]	[60]	
	[18]					[2]			[41]	[48]		[54]	[61]	
	[19]					[26]			[42]			[55]	[62]	
	[20]								[43]			[56]		
	[21]													
	[22]													
	[23]													
	[24]													
	[25]													

and user values were clearly defined. This resulted in increased usability and product owner satisfaction.

Wireframes were used in [24] as part of the process for redesigning their current website, providing a concept for the layout, structure, and navigation and organizing content and prioritizing content. The validated wireframes were turned into a mockup. Using case studies, the authors of [25] suggested employing scenario techniques in Extreme Programming to demonstrate how usability techniques impact agile software processes. They conclude that using Scenarios may involve and get better results from users because this method focuses on the user's needs and not on the functional software as is in the user stories. Therefore, they conclude that using scenarios produces better results because they focus on users' needs rather than solely on the functionality of the software like user stories.

3) **Personas:** In [21], the authors have used personas to develop a detailed description of the planned application as three primary personas have been emphasized, which resulted in the successful implementation of the application in accordance with its scope and methodology. According to [26], within U-Scrum, the product owner collects and consolidates data from stakeholders to create a small set of personas. During meetings with the development team, the personas are discussed, and posters describing them are placed throughout the work area. Developers and product owners use personas to demonstrate their points, such as when considering the viability of a new user interface.

4) **Workshops:** A collaborative workshop called Design Studio was used in [27] to demonstrate the benefits of incorporating agile into the UCD process. Developers and designers attend design studios to discuss their work, ideas, and feedback in order to have a common understanding. In [3], used workshops as a normal method of communicating with the team, initiated by the product manager. Workshops are designed to help product managers understand how to handle the development process in an agile environment.

5) **Usability testing:** A formative usability testing was used in [28] to propose an integrative approach that integrates the agile methodology and the UCD to create real-world social robot applications. The usability testing used to evaluate each system iteration with variety of participants to eliminates usability problems early on and its focus on produce useable prototype with a high quality. In [29], the researchers proposed an agile and comprehensive method that applied traditional desktop eye-tracking techniques for qualitative and quantitative mobile usability testing. The authors propose two hypotheses (H1 and H2) that relate to two eye-tracking metrics: (H1) a method for integrating desktop eye trackers with real-life usage of mobile devices, and (H2) recommendations on how to implement the concept. In terms of agility, this method is highly integrated into the process of UCD (User-Centered Design). It can be applied at the early stages of development. While the method still requires further testing, it represents a good start towards a Comprehensive and Agile methodology for quantitative mobile usability testing, which offers numerous benefits.

In [30], the researchers proposed adopting agile software development methodologies and user-centered design approaches to application development that emphasize iterative user interface development involving usability engineers and non-technical users. The approach aims to apply usability testing procedures to mass-market applications on mobile phones. The application is designed to meet users' needs, with a focus on usability and customer satisfaction. As part of the system development life cycle, usability engineers assist the development team. The engineers provide suggestions that are continuously integrated into the system. Moreover, test users are also included in the process, resulting in additional benefits. As a result of this continuous input, the system can be adjusted to effectively meet the user's needs.

6) **Lean UX:** In [31], the authors described how to integrate the lean UX process with Agile processes by having different sprints put under different themes where each theme might span many sprints. The theme begins by sketching ideas to get a skeleton of design output to be used in the next sprints. Each sprint lasts for two weeks, and users validate the work at the end of every week, and the feedback is used to revise the design. During each sprint, MVPs should be produced, and the most important features should be prioritized so that user testing can be focused on the core. Then they mentioned that Lean UX philosophy is a combination of three influences: design thinking movement, Lean start-up method, and Agile development.

The authors in [19] combined Scrum with the Lead UX philosophy, which has three essential components: design thinking, Lean manufacturing, and Agile development. After each iteration, the user and developer used interviews to get feedback to resolve the most major issues: user control and independence, user assistance, error recovery, and programming skills. Furthermore, the most significant findings were in the areas of aesthetics, minimalism, and the use of language.

The researchers of [32] investigated how the lean user research is done in agile methodology and how the DACH region's UX community is present in the industry field. Thus, the authors provide a Lean User Research pattern. The proposed pattern describes the best practices regarding user research in the agile environment to allow the agile team to improve their understanding of users. Therefore, the proposed pattern includes short feedback cycles, which helps the UX design for fast action regarding changing requirements.

In [33], the article described how a large multinational telecommunications corporation used lean transformation to rearrange software development activities. The UX designer's work is integrated with other software design methods during the transformation phase. By analyzing documentation and conducting semi-structured interviews with seven software development professionals, the new agile style has raised UX to a central concept for planning, creating, monitoring, executing, verifying, and validating development processes. As a result, lean transformation and agile software development approach successfully integrated UX and UX-related documentation at the center of planning and goal setting. UX design had become very adaptable.

3.2 Classification of Articles Based on Techniques

4) **Workshops:** A collaborative workshop called Design Studio was used in [27] to demonstrate the benefits of incorporating agile into the UCD process. Developers and designers attend design studios to discuss their work, ideas, and feedback in order to have a common understanding. In [3], used workshops as a normal method of communicating with the team, initiated by the product manager. Workshops are designed to help product managers understand how to handle the development process in an agile environment.

1) **Scrum methodology:** Several articles have adopted UCD techniques in scrum approach. In [34], the authors adopted Storytelling and prototypes into scrum approach to develop an assistive technology which enabled the user advocate to be more involved in exploring the users' needs without delaying the development of the software as well as the development team was aware of the usability without spending copious amounts of time thinking about it. The authors stated that this approach is effective for designing systems for and with hard-to-reach populations. The authors in [35] combined the scrum approach with a formative user experience evaluation to improve the design instead of drawing conclusions about the system. They indicated that this integration ensures the project remains organized and emphasized the creation of a usable product.

In The Development of One Stop Service Online System based on User Experience Design and AGILE Method [36], the aim was to design and develop a One-Stop Service system for Educational Media and Technology Services, which consisted of two phases, the first of which involved designing and developing the system in accordance with user experience and using agile methodology, while the second phase involved evaluating its efficiency and usability. Scrum technique was chosen as agile methodology, which involves repeating the process for each incremental change over an iterative process. It started with the integrated UX and Agile processes, which were present in UX:Agile's output line: 1) understand, research: sprint planning, 2) sketch: design, 3) design: develop, 4) implement: test, and 5) evaluation: review.

Another study [16] presented a scrum-based development approach that incorporated many characteristics of user-centered design, such as the involvement of experts and users in the analysis, design, and implementation as well as paying special attention to their feedbacks toward additional changes and features in subsequent sprints. The result demonstrates that when applying this process, the interactive design is improved to design comprehensive educational programs. In [2], the authors applied the scrum approach in two cases. In the first case, they used scrum to update and modify the previous version of the product. While in the other case, they used scrum to develop a new project for a new client. In addition, they employed SCRUM to fit the call for tendering document (CFT) and the project contract.

The authors of [26] proposed U-SCRUM as a variation of SCRUM. In contrast to typical Scrum, where only one member handles usability, U-SCRUM has two product owners, one focused on usability and the other with a more detailed approach to the project. Based on preliminary findings, the authors concluded that U-SCRUM produces better usability than SCRUM.

2) **Extreme Programming (XP) methodology:** UCD has been incorporated into Extreme Programming (XP) approaches by several articles. The authors in [14] proposed an approach for integrating UCD and XP development by evaluating the usability of user interfaces as applications are developed in small iterations. Their study used various HCI techniques, such as user studies, personas, usability expert evaluations, usability tests, and lightweight prototypes to evaluate the usability of user interfaces. The authors claimed that integrating both processes make it possible to combine the benefits and minimize the failures of each, as XP needs to know its right end-users, and integrating with UCD provides the answers to these questions.

The authors in [21] proposed an approach for integrating UCD and XP development by adopting prototypes to develop a Halal-Checker mobile application. As a result, the Halal-Checker Mobile Application was successfully aligned with the project scope and methodology. In addition, users could access the applications without assistance or instructions, and the feedback from the prototype testing was consistent with the simulated user scenarios.

3) **Feature-driven Development methodology:** A single artic has incorporated UCD methods into feature-driven development approaches. In [37], the researchers proposed combining UCD and FDD to present first impressions of the look and feel and acquire the hands-on experience of prospective community members at an early stage. Their integrated UCD and FDD processes have developed a system that can be considered a general tool for users across different contexts. The results showed that the proposed approach helped iteratively create a product that promptly addresses the users’ needs and requirements and eliminates any defects.

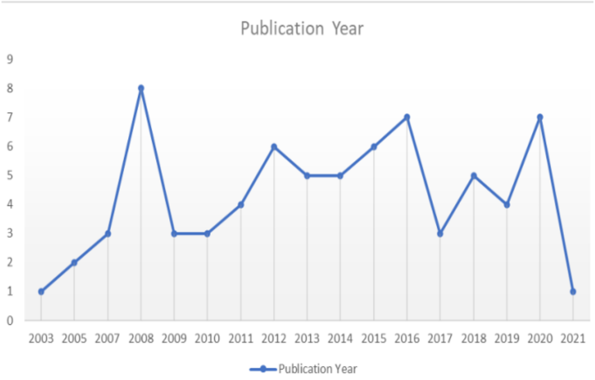


Figure 6. Publication year for the selected studies

4) **Parallel track:** A parallel track approach was used in [10]. In each iteration cycle, designers and developers worked together seamlessly to gather requirements and design for the next cycle while testing the previous iteration's work. In [35], the author used a parallel track development model in an undergraduate summer research project that recommends conducting UX testing a Sprint before the implementation. This model helped them efficiently combine design and implementation. Despite that, they wasted some time at the end of the summer when they did not get to implement a feature that they carefully planned of time.

In [38], the aim was to collaborate user experience and development teams in a cross-functional way. The process consists of two tracks: Discovery and Delivery. Both tracks run continuously and in parallel, with Delivery being a part of Discovery, which helps the development team participate in discovery activities to demonstrate the competitiveness of the new product ideas and their technical feasibility. The result demonstrates that the team can deliver usable and useful products that satisfy the user's needs when applying this process.

In [39], a parallel track was set up, where the UX scrum team and the development scrum team collaborated at every iteration. This has resulted in complete and more accurate UX stories, allowing UX teams to devote more time to their work rather than hunting for requirements. In addition, this projected capacity made it easier to balance additional tasks like bug repairs and scope adjustments. The authors of [40] used UX integration into Scrum practice to work parallel within a sprint. Developers and UX designer teams work on parallel tracks to keep their activities synchronized. This results in benefits such as designers focusing more on exceptions instead of striving to get the best design right the first time, better design planning, and more accurate quotes for customer satisfaction.

In [41], the author described Alias' efforts to integrate agile development and User-Centered Design processes by building parallel design and development tracks. Alias has begun work on a new product, Alias® SketchBookTM Pro. The SketchBook Pro team decided to use agile development principles for their first release. The usability team members used this as a chance to modify their customer input process to make it more compatible with the new development model. The usability and development adjustments were a massive success, and subsequent SketchBook Pro releases have been built in the same way.

The author of [42] realized that understanding how Agile developers and User Experience (UX) designers collaborate in the real world is essential. The work of developers and designers is influenced by the organizations' ideals where they work. In their study, two points emerged: (1) separating Agile developers, and UX designers are the most effective way to develop quality software, and (2) Agile developers and UX designers working closely together is the most effective way to develop quality software.

Agile UX Integration Practices and UX Vision were two themes explored in this study [43] to establish a framework for integrating UX in Agile. This approach is designed for lean UX work to be done in Sprint 0, while Sprint 1 has an additional UX layer on top of the typical sprint layer, allowing the team to work in parallel while also ensuring that everyone is on the same level about what has to be done in the sprint.

5) **Hybrid methodology** Several articles have proposed different models that integrate Agile and UX/UCD. The researchers in [44] integrated UCD into Scrumban, a hybrid method of both the Scrum and Kanban agile methodologies for better and faster usability design. Scrumban reduces the team workload by breaking down tasks and assigning responsibility. In the proposed method, the UI owner is added as a new team responsible for UI design assessment. Combining Scrumban with UCD benefits rapid software development since time is strictly managed and UI design constantly evolves. The authors of [45] employed an agile usability engineering methodology named InterMod in the development process of a real mobile application project. During the early stages of development, questionnaires, interviews, observations, thinking aloud, and usability testing of paper prototypes are used to manage user expectations and discover usability issues; therefore, accurate decisions can be made.

InterMod has proven its effectiveness through the proper use of time and effort. A similar InterMod approach was presented in [46]. The authors created high-quality interactive applications by combining Agile Methods, Model-Driven Development, and User-Centered Design. They proposed combining three philosophies: AM, UCD, and MDD in their presentation. Following Agile Methods, the User Objectives to be addressed are iteratively defined throughout the process. As a result of this strategy, two outcomes were achieved: (1) flexibility in planning and dealing with changes that occurred during the project life cycle as a result of "user wishes"; and (2) earlyvalidation of the models included in SE-HCI allowed the UOs to progress and lead the subsequent activities.

An agile hybrid approach was proposed in [47] for producing highly usable qualitative web applications based on defect quality analysis during the development stages. Agile web development and lightweight UCD practices are applied in this approach, including user research upfront, iterative design by the UX developer, and usability testing throughout the design process. The authors empirically evaluated how the proposal affected quality in a university case study. Two separate projects have been assigned to different teams to collect quantitative data (Open and Closed Defect data records) related to user stories, implementation, and testing activities across all development phases. This method results in a better product by reducing defects and improving the distribution of closing defects over the sprints.

In [48], the authors propose a UCD-Agile hybrid approach based on User Objectives (UO) to collect and develop requirements. User requirements are collected in UOs by gathering functional and non-functional requests from the user. UO completion entails three activities: Requirement specification, Presentation, and Functionality. This allows for flexibility in planning tasks in accordance with agile and UCD. UOs are classified according to two factors: their development and their relationship with the user. With UOs diagrams, the project's progress can be visualized, which facilitates managing priorities, prioritization, and planning of the project and gaining insight into the UOs that have already been evaluated and those that need to be evaluated.

3.3 Classification of Articles Based on Practices

Several articles have proposed different models that integrate Agile and UX/UCD. The authors in [49] proposed a practice-oriented framework that utilizes Pair Programming as an agile practice and UI design practices. The framework was taken to apply to CS students. The author also mentioned that the framework could be extended to involve industry teams since it has increased developer and user satisfaction. Therefore, it focuses more on user experience (UX) components while developing the software.

User stories are used in [39], which was written by the product owner (PO), who is responsible for defining the stories and prioritizing them. The UX team employs prototype techniques to confirm if stories fulfill the requirement. In [40], product backlogs list a bestselling product's most important features or capabilities. When a priority feature is added to the product backlog, the team will first focus on the most important or highest priority item.

3.4 Classification of Articles Based on Different Approaches

1) **New Framework:** The use of Agile and UX/UCD in different frameworks has been proposed in various articles. In [50], the authors proposed a novel framework that integrates the agile approach with the user-centered process to bridge the gap. They applied their work to a disaster management Earthquake case study that involved different users with a significant role in the crisis. And, since crisis management systems should be heterogeneous, the service-oriented architecture (SOA) was chosen as the supporting architecture. Their framework consists of four main phases (i) Organization and agile business analysis, (ii) Just in Time (JIT) Requirements Analysis and Elicitation, (iii) Iterations Prioritization and Planning, and (vi) Release to Iteration. The researchers of [51] introduced the three communication breakdowns in integrating UCD and Agile that evolved from the literature and previous work of their Smart Campus project. These breakdowns include contradictory thoughts on user involvement, documentation, and iteration. They improved these themes by analyzing a case study of an IT company that defines a communication network and artifact types used, such as Mockups and Wireframes, Briefs, Roadmaps, and Technical Analysis. The researchers extended the framework with a task ownership theme and argued that communication breakdowns could be focal points for action and decision. They also suggested adopting design thinking and customer engagement.

In [52], the authors proposed a three-fold integration framework that adopts and applies the UCD philosophy from end to end at all levels to fill the gap that agile development teams often lack in integrated development. Their framework consists of three levels: life-cycle level, iteration level, and development-environment level. Their approach identifies best practices for applying UCD alongside agile development processes and gives suggestions and recommendations for incorporating UCD into agile software development at different levels. A framework called SIBAP (Script-Based Aspect-Oriented GUI Prototyping) was presented in [53] that reuses prototypes early on until they become the final product. Designers and developers share the same prototype to reduce the disconnect between them and guide their efforts toward the final product. The study indicated that the framework is effective and benefits from a faster development process.

The authors in [54] proposed a framework that integrates scrum and user experience design techniques into the software development cycle, along with the Capability Maturity Model of Integration (CMMI) standards and the dimensions described in the Human Factors Institute's maturity model for user experience design. The study indicated that the framework would improve software development. The researchers of [55] proposed an Expert System that can assist in developing interfaces during Agile Software Development. This system provides the designers with the information they need to make their best decisions, particularly during development. This procedure is built into Agile Methodologies like scrum, where it was discovered that good practices in constructing human-computer interfaces had been abandoned in favor of decreasing sprint times.

In [56], the authors proposed a framework to integrate scrum and UCD practices by keeping the usability principles and practices in mind. The goal is to improve the usability of the developed product, minimize the costs, risks, scheduling, and late changes, and minimize the impact on the Agile development process. The framework's three primary aspects are user/client satisfaction, fewer changing requirements, and communication between the design and development teams. The results from the framework can be applied to collocated development, provided they meet conditions that apply to the distributed environment, and it is mainly applied to websites and mobile applications.

2) **New Model:** Several articles have proposed new models integrating Agile and UX/UCD. The researchers of [57] introduced a new combined model that reduces the time it takes for UCDs to communicate their knowledge to the software development team. The model contains four stages. Requirement analysis and user study are the first stages where a workshop is conducted between designers and developers' teams to resolve understanding conflicts by presenting slides to discuss the new ideas in the initial stages. In the second stage, a prototype of the system was developed and evaluated with usability testing. Once a newly developed version of the system is ready for evaluation, a meeting is to discuss usability issues, suggest modifications, and provide feedback. Finally, the system is integrated, incrementally built, and evaluated using an ongoing usability test.

An extended sprint model with a limited duration is proposed in [58] that integrates Scrum development with UX workflow. The design sprint one outputs are used in development sprint 1, where the design team moves on to sprint 2 to work on the next set of features. In addition, designers are involved at the end of the process to test the final product and ensure that they meet established usability standards. Using this model, the author asserts time and quality issues can be addressed. A conceptual model for the UX-Scrum integration was presented in [40], which showed how UX designers work within Scrum teams, using three approaches: parallel working, working within a sprint, and lean UX. In parallel track approaches, development and UX design teams work in parallel tracks simultaneously to ensure their activities remain synchronized. The second approach focuses on the collaboration between UX and development teams during a sprint. UX designers can work with product owners or development teams, depending on which sprint is taking place. During a Lean UX approach, the team will be more focused on the outcome than output, which generally refers to a product called a Minimal Viable Product (MVP).

A maturity model was proposed in [59] that adopts XP, Lean, and UCD to overcome the gap on how to identify whether a team is maturing through transformation. In addition, they aimed to define a method for assessing the team's maturity at a particular point. The result demonstrated that their model differs from the literature on several points: (i) All decisions are based on experiences, regardless of the expert’s intuition. (ii)The team inspires itself and participates in decision-making without the presence of a leader. (iii) An anchor role links the understanding between business and engineering. (iv) Teams constantly strive to use the latest technologies that benefit their cause without using all UCD techniques. The initial contribution could already be useful to software development teams striving to adapt to such changes.

A process model was proposed in [60] that illustrates how two industry teams employ a combined Agile, Lean Startup, and UCD approach to software development. The teams defined three phases for the entire development process, in order: scoping, which aims to determine the scope of their work; discovery and framing, which entails refining the problem to solve and then determining the best solution to solve it; and iteration, which entails continuously developing and properly implementing the chosen solution. The authors stated that this model could be utilized as a starting point for enterprises to create their customized integrated strategy.

The authors of [61] presented a process model that integrates agile software development with user-centered design and usability techniques. The process model recommends having a common phase where developers and UI specialists participate in activities. In addition, this will facilitate the early detection of design usability problems and design implementation, resulting in a less time-consuming development process, resulting in a rapid development process and more compatibility with customer needs later.

The authors of [62] proposed a maturity model for integrating Agile processes and user-centered design. The model provides an assessment tool to determine an organization's ability to integrate Agile processes and UCD. The AUCDI maturity model comprises three elements: First, a multidimensional reference model that identifies the fundamental elements that affect AUCDI. Second, a performance scale for assessing the organization's or project's performance on each element assessed in the AUCDI reference model. The third is an assessment procedure that provides practical guidance on performing the assessment. The model helps developers better understand usability and UCD and gives UCD practitioners a way of pinpointing areas that require improvement in usability processes and practices.

3) **New Team role:** A new role for a team called "On-site user experience consultants" is proposed in [63], [64]. In [63], the researchers presented their experiences integrating UCD into agile RE in fixed-price projects by introducing an On-site User Experience Consultant role. There are three phases of improving UCD activities: First, UCD during initiation includes all the activities that must be completed before the development process begins. Second, osUX consultants provide feedback to developers on preliminary results during implementation to ensure that users are represented in review sessions. Finally, UCD's follow-up phase can help check whether requirements have been met and whether the solution provides an excellent user experience.

The researchers in [64] focused on fixed-price projects that integrate user-centered design (UCD) in agile requirements (RE). Their solution allows customers to change requirements without extra payment. They also introduced a team role called “On-site User Experience Consultant” (osUX consultant) for a UCD-trained staff to work with the users’ needs and provide usability consulting during the development process. They used personas and scenarios in requirements workshops and ramp-up meetings before implementation. In addition, they suggested changing communication habits.

The authors of [65] recommend that agile projects require more UCD resources by distributing the UCD teams' responsibilities into two team roles: a UCD researcher and a UCD prototyper. The UCD researcher has many responsibilities, such as gathering feedback, while the UCD prototype focuses on the UI prototype and its characteristics. However, they work closely with each other. In addition, the researchers suggest that the activities related to UI design should be at the initial stages of the development process.

4. RESULTS

4.1 What are the most used UCD practices in agile development? (RQ1)

Various UCD techniques have been employed in Agile development. For instance, Little Design Up Front techniques have been implemented in several articles by reserving a cycle zero, or sprint zero, for analysis and design before any actual iterations start. This concept revolves around time-boxing the analysis and design phases for the duration of the development cycle. Persona was used in [10] at the beginning of the project, particularly in cycle zero. A pre-development usability evaluation was conducted in [12]during Sprint 0, while [11] described how a one-day design studio fuses agile development team practice with user-centered design. A three-fold integration framework was proposed in [52] to adopt and apply UCD from end to end at all levels.

A collaborative workshop called Design Studio was used in [27] to demonstrate the benefits of incorporating agile into the UCD process. However, [3] used workshops where the product manager initiates them for everyday contact with the team. Additionally, [3] used workshops as a normal method of communication with the team. Fixed-price projects in [64] integrated user-cantered design (UCD) in agile requirements (RE) and introduced a team role called “On-site User Experience Consultant” (osUX consultant), used personas and scenarios in requirements workshops. The researchers of [57] introduced a new combined model that reduces the time it takes for UCDs to communicate their knowledge to the software development team.

4.2 What are the agile methods that adopt UCD techniques? (RQ2)

Based on the literature, Scrum and XP are the most common agile methods in combination with UCD techniques, where they provide more flexibility in software Iteration than other methodologies. Also, the software team is more familiar with such methodologies. In addition, some articles combine other agile methodologies with UCD techniques such as FDD, InterMod, and Scrmban. However, there were a lot of articles that proposed different frameworks and models. As an example [50], a novel framework based on integrating agile processes with user-centered approaches was proposed to bridge the gap between them. Another study [51] has proposed three communication breakdowns that evolved from the literature and previous work of their Smart Campus project.

In [52], the authors proposed a three-fold integration framework to fill the gap that agile development teams often lack in integrated development. Another study [53] presented a framework called SIBAP that aims to enhance the development process. [54] proposed a framework that adopts Capability Maturity Model of Integration (CMMI) standards to improve the development process. In [57], the authors presented a combined model to reduce the time it takes the UCDs to communicate with the software development team. In [58], a limited-duration sprint model is proposed, integrating Scrum development with user experience work. As a result of this model, the author claims both time and quality issues can be addressed.

A conceptual model for UX-Scrum integration has been presented in [40], which shows how UX designers work in Scrum teams using three approaches: parallel working, sprinting, and lean UX. A maturity model based on XP, Lean, and UCD was proposed in [59] in order to overcome the challenge of identifying whether a team is maturing through transformation. A process model was resented in [60] that outlines how Agile, Lean Startup, and UCD approaches work when applied to software development projects. Using this model, enterprises can develop their customized integrated strategy. [39] presents an integrated process model for agile software development, user-centered design, and usability. This model enables early detection of usability problems and design implementation, resulting in a more rapid development process and greater compatibility with customer needs. Two themes for lean UX were presented in [43]: Agile UX Integration Practices and UX Vision. The themes help the team to work in parallel and to have a full awareness of work progress. Moreover, the themes used in Sprint 0 and Sprint 1. In [48], the researcher proposes a UCD-Agile hybrid approach based on User Objectives (UO) to provide flexibility in agile-UCD task planning.

4.3 How does integrating the agile method and UX/UCD improve the process? (RQ3)

Agile and UX/UCD integration have led to improvements in several articles. The researchers of [66] conducted interviews with Agile teams that use User-Centered Design (UCD) and found that Agile enables usability testing on working software and allows to detect and correct usability issues later during the iteration process. In [67] stated that including the User Experience team and incorporating UED principles into the Agile development process greatly outweighs any risk of derailing product delivery. The authors highlighted in [14] that integrating users into the process indirectly improved it through HCI instruments, co-location, communication, and planning meetings and increased the morale of their entire team.

The researchers of [44] showed that utilizing UCD within the Scrum ban allows for more efficient and faster software development and usability design. The authors of [68] demonstrated that discount usability testing that integrates with Scrum could be rapidly accelerated by utilizing low-cost usability evaluations that do not require as much time or effort as complex statistical methods. As part of the discount usability approach, prototypes, heuristic evaluations, and simplified think-aloud protocols were used to get early feedback from the users. The authors of [48] claimed that the combination of agile integration and user-centric design (UCD) provides the advantages of both approaches. It is expected that requirements will gradually change and adapt to meet the needs of customers and users and that UCD techniques will make the software more usable and improve the User Experience (UX).

The findings of [54] revealed that SCRUM facilitates the improvement of a process by providing a better understanding and management of the process. The improvement process can be achieved in a sprint or several sprints as each iteration is composed of five steps: planning, designing, executing, presenting the improved process, and presenting the next iteration. As reported in [2], the integrated process between UCD and Scrum demonstrates improvements in the development process, such as reducing reworks, satisfying users, and better collaboration with stakeholders, along with a better understanding of users and their needs. In addition, the integration provides some additional techniques that can be used in other contexts. In [40], it was found that parallel tracks had some benefits, such as designers tend to focus more on exceptions rather than striving to get everything right on the first attempt, better design planning, and better customer satisfaction quoting. Several of the development activities that were featured in [69] are integrated into multiple firms' agile process efforts, with many of the activities including user experience design (UED). The author found that teams had made great progress in the Internet Start-Up, Financial Services Provider, and On-Line Service Provider.

The authors of [70] proposed a requirements analysis method that bridges the gap between user tasks and application features without degradation of characteristics of Agile methods and UCD practices. It consists of a lightweight requirements specification method in Agile-UCD that encompasses three essential aspects: Streamlining communication and collaboration between UX designers and application developers while simultaneously developing application designs. A user and application system workflow based on a user interface uses compatible modeling codes to develop visual representations. Use workflow-oriented usability criteria to determine whether functionalities are usable in the object-oriented model. Although the method successfully improved the main issues of Agile-UCD in the requirement specification phase, there is still a need for modeling training to make the approach work in practice.

The author in [71] emphasized the importance of collaboration between designer and developer, particularly at the beginning of project work, a stressful and impactful phase. This strategy is especially important so that the entire team will better understand the program and generate suggestions on how it can be improved from the end-user, business, and technological standpoint. It entails creating software collaboratively and constructively at the start of a project. The benefit of collaboration is focusing on getting this work done skillfully instead of attempting to resolve multidisciplinary issues.

A logging analysis approach is proposed in [72] for integrating UCD practices into agile processes. The purpose of this study was to examine the usefulness of log file analysis to enhance the match between the health care system and its users through agile development. Agile professionals can gain valuable insight into users' behavior by analyzing log files. Using the results, professionals can better understand uses' behaviors and make the methods less labor-intensive.

In [73], the researchers presented a case study about UX work in Agile software development. The study aims to improve UX work in Agile software development projects. The Agile UX process results in a systematically effective user experience. Findings suggest that the UX team needs to be skilled, but management decisions and power relations play an essential role in determining the impact of the UX work.

In [74], Designers of User Experience (UX) and Agile developers have concentrated on bringing the disciplines together by integrating procedures or adopting methodologies with their interactions. The author claimed agile development approaches provide high-quality software faster, and interface design met our end-user satisfaction goal. It also provides the ability to give feedback on UX design at regular intervals.

4.4 What are the challenges of the integration of agile and UX/UCD? (RQ4)

Based on the literature, challenges related to agile integration and UX/UCD were categorized into six categories.

1) **Time:** According to [68], Agile's time-boxed nature poses challenges in scheduling and conducting usability tests. For instance, [14] stated that usability tests could not be conducted frequently with end-users. Also, [65] point out that agile methodology requires a fast development process which in this case affects the time spent for the complete understanding of the customer requirements; thus, by distributing the workload in the UCD team, it should be able to overcome this challenge. Lack of allocated time for upfront activities was addressed via upfront design, as in [10], which used persona at the beginning of the project, particularly in cycle zero, helped in planning and gathering customers' data. Another study [75] declared that UCD spends a lot of time and effort researching and analyzing before developing.

Another problem mentioned in [35] is the open-ended nature of design tasks; it is sometimes hard to determine when an idea is finished. As described by the author, they allocated time for inexperienced designers to spend on design. Once that time ran out, they deferred further decisions till the evaluation. Another study [11] used a one-day design studio that helped the entire team understand the design better, allowing them to begin development and improve collaboration. In [27], the researcher chose to use a collaborative workshop, a design studio that provides designers, developers, and stakeholders with a quick way to try out different design options. Consequently, designers and developers can improve their skills by freely discussing their work's merits and shortcomings.

Another solution is using usability engineering techniques, a practical way to conduct low-cost usability evaluations that do not require as much time or effort as traditional methods. For instance, an expert evaluation was applied in [14] that helped to mitigate frequent usability tests. It also has been suggested in [45] that using usability evaluation techniques can reduce development time and effort. It also avoids the cost and effort of reworking incorrect paths.

2) **Work Balance:** Researchers of [74] observed that the issue of Agile Developers and Interaction Designers having to work on a software development project has historically been characterized as a problem of integration. Another issue appeared in [39], where UX teams often struggled to get input from development team members since they worked before scrum teams. Due to the user's requirements and roadmaps being unclear or poorly defined, the UX team spent a lot of time gathering new requirements. As a result, new requirements were added, while other problems such as bug fixes appeared. The researchers in [75] mentioned that UX designers in agile teams are not often considered full-time staff, and some organizations have a shortage of UX designers involved in agile projects. Due to this, UX designers are constantly working on multiple projects and becoming overwhelmed. According to [63], UX designers cannot collaborate closely with developers because they are working on several projects simultaneously and are too busy with several projects.

To overcome this challenge, several techniques were suggested. In [66], the authors demonstrated that screen mockups could improve relationships between software developers and user interface designers. Similarly, another study [76] mentioned that artifacts such as sketches, lists, and stories are vital to collaboration between Agile developers and UX designers. Also, several practices were suggested to overcome this challenge; for instance, a design studio was suggested in [27], which facilitates a two-way knowledge transfer as developers can gain more insights into design while designers gain a better understanding of how developers work and how their challenges may affect design.

Another solution was presented in [77], where the authors focused on training the software developers to do UX work. Accordingly, software developers can perform the modified AB usability test within sprints. Their feedback is considered for enhancing the training to improve the usability and UX design processes in agile methodology. This approach ensures its quality where software developers can conduct usability tests independently of UX designers. In addition, the researchers of [78] have shown that successful integrating Agile and UX work depends on attitudes and practices, such as mutual awareness, expectations of acceptable behavior, negotiation of progress, and general engagement. In [39], the researchers proposed utilizing a prototype to verify that the user's requirements are well understood. Moreover, create a collaborative team between UX scrum and development scrum to work parallel in each iteration.

To integrate UCD operations in distributed agile teams, two approaches were used in [79], informal and formal. Formal methods employed agile project management tools such as Jira and Rally. The informal communication methods can be categorized as either synchronous (e.g., video conferencing, telephone conferences, or chat rooms) or asynchronous (e.g., emails and forums). The author discovered that integrating UCD activities into distributed agile systems required coordination between UX-design and development teams.

Another challenge was mentioned in [5], as agile focused on workable software while ignoring usability and the end user's involvement during the development phase. On the other hand, the usability approach focuses on the user's needs to develop usable software. The authors attempted to resolve the trade-off between usability and an agile approach by providing a developed model that involves a six-step process, each of which is used for the following.

3) **Prioritization:** Since developers primarily focus on completing functionality rather than usability or UX features, it would not be easy to prioritize usability and UX-related activities within the different sprints and iterations [26]. The authors in [26] overcame this challenge by using the U-SCRUM technique specifically to help improve usability. This

Table 2. Future Directions

Year	Paper Ref. No	Proposal for future Articles
2021	[28]	Future work involves improving the performance of the proposed approach by using summative usability testing. In addition, more research could be done on merging summative and formative usability testing into the agile development approach.
2020	[60]	For future work, since the teams likely used different approaches to represent their work and likely used a structured, activity-centered approach, instructing the teams on how to represent their activity-centered work process in an unstructured manner would be an interesting exercise.
2020	[3]	Future research will focus on realizing how other activities are affected by deliberation and concise user stories in agile software development. In addition, more studies are needed to explore how to reinforce practices for integrating agile UX processes on different organizational levels.
2017	[44]	The proposed future work will extend the Scrumban method to be applied in large-scale systems with an appropriate board.
2016	[3]	More empirical studies in academia and industry can provide more reliable evidence supporting the findings, and another type of case study will be established with different web projects.
2016	[46]	The future work for developing the User Objectives, distributing tasks in parallel speeds up resolution, enabling progressive gathering and validation of requirements.
2016	[47]	Further empirical studies in academia and industry can provide better reliable evidence to support the findings, along with another case study of the same type with different web projects.
2016	[53]	The authors suggested future work on improving the SIBAP framework to resolve limitations and utilize different scripting languages' features. Also, they recommended conducting further experiments to determine SIBAP's effectiveness.
2015	[18]	In future work, a comparison between the usability of systems developed by integrating Agile UCD and the traditional approach that incorporates usability tests.
2015	[19]	The future work will involve validating the system in other areas of SEN children after the program is improved.
2013	[34]	For future work, the authors suggest investigating the scrum methodology with assistive technology and gaining new insights from end users in the HCI domain.
2012	[29]	Further work is planned on mobile usability testing, including exploring new methodologies. The proposed method applies to new devices, such as tablets. On the other hand, this approach is intended to be applied to implementing field tests in real-life situations.
2012	[78]	The future work entails exploring more Agile/UX tools, processes, and methods. In addition, investigate how these processes and tools collaborate in social processes. Moreover, the authors suggested analyzing practice in terms of culture.
2010	[57]	In future work, the Inter-Combined Model will be used in further projects to evaluate its validity in bridging the gap between UCD designers and software developers.
2010	[17]	For future work, a bridge will be built between the prototype and the application features being developed to bridge the gap between recording tests on a low-fidelity prototype and running them on a separate GUI.
2008	[11]	In future work, the authors suggest applying the design studio approach to enhance practitioner adoption of UCD.
2008	[76]	Future research will focus on implementing a more comprehensive model and examining other specific collaborations in greater depth.

Table 3. Articles Limitations

Year	Paper Ref. No	Articles Limitations
2020	[60]	The methodology had flaws since the researchers used numerous data sources to verify their conclusions, and they had senior researchers follow each step of the investigation to mitigate construction validity issues. In addition, the model cannot be claimed to be valid in other contexts in terms of external validity. They used automated digital technologies to check their modeling efforts' syntactic quality. Despite that, the semantic quality was improved due to the supervision and modification of the scholars and experts mentioned.
2020	[72]	As the data for the present study were collected at a large-scale software company developing health care software, there are some limitations. No data was collected on small or large-scale companies with fewer than ten staff members.
2019	[3]	This method limits study control and does not enhance the generalizability and impartiality typically associated with scientific knowledge.
2016	[53]	SIBAP framework is limited in its use of scripting languages, where prototype behavior is affected by languages that lack features.
2016	[47]	The limitations of this research include issues with the total time of the study (about three months for six sprints, including sprint 0). Also, it threatens to move students from theory to practice even after training. As a Proxy role for a UX/UI developer, a student is assigned, but in the industry, the designer should be a separate experienced developer. Some defects were corrected quickly without recording; others were not recorded. Case studies often cannot give accurate results; using various empirical analyses with different web projects could give different results.
2015	[77]	The finding limits the scope of usability and UX methodologies appropriate for the proposed approach since the developers faced some issues in the data analysis.
2015	[18]	The limitations of this study are that they did not follow a complete development process. So, the need for usability testing is not confirmed even though no complaints regarding the product's usability were received. They were generalizing their findings, and the analyzed studies were considered odd since they do not cover all the possibilities, and the contexts might differ.
2013	[34]	This approach has disadvantages in describing how a participant used the software and the difficulties with the development team. As participants have different physical impairments, they will reach the software differently. Video feedback greatly benefited the user when providing feedback to the rest of the team. Another disadvantage is relying on one individual to go to the care center, which posed problems since the individual could not attend due to physical limitations and transportation problems.
2012	[78]	The findings of this research can be applied if generalized care is needed, and the study does not cover all options. Agile development is much more than Scrum, and the way designers follow it differs from practice. However, the teams studied are regular teams in practice.
2012	[35]	The researchers had to adjust their Scrum methodology due to the lack of an official owner and the time limitations. There were continuous plan changes, some Sprints remained incomplete, one Sprint was dropped after the task estimation was incorrect, and one-week Sprints reduced errors.
2010	[17]	There are limitations to doing repeated usability evaluations of a prototype of a GUI. The number of changes developers will have to make later in the development process will be minimized.
2008	[14]	The usability engineers were not physically present all the time, which can help in solving design issues perfectly if done correctly. Additionally, due to time constraints and budget constraints, they did not conduct a lot of usability tests with users. However, this problem was alleviated through the usability expert evaluations.
2008	[25]	A significant issue they observed in the academic environment was the work required to compose good scenarios. Additionally, it was challenging to incorporate technological constraints into the situations as they appeared to be unknown and could only be tested using software prototypes. Finally, technical information may be required. In an attempt to catch the attention of their creators, they introduced the role of the scenario champion, a person who attempts to raise awareness about the situation.

methodology improved their project, undertook in-depth investigation and analysis, and created a product concept.

Another problem mentioned in [41], during the initial iteration planning meeting, issue iteration objectives were defined, and the product team roughed out which features would be in which iteration. (Each cycle was about two weeks long). The researchers in [41] overcame this challenge by using the priority data to ensure that they first worked on the most important features. This ensured that if something had to be dropped, it would be one of the lower-priority features.

The researcher of [27] reported that the design studio provides a shared design vision with software developers and allows them to engage in usability practices, minimizing rework and integration issues in the earlier stages.

4) **Feedback:** A key advantage of agile, specifically XP, is that the customer can provide feedback to the development team. The customer is usually an expert in the functional area covered by the program. However, it would be challenging for that user to anticipate potential usability concerns for different groups. Also, users may have very different mental models about the area being addressed by the software, so getting feedback from a larger group of users about potential usability concerns may be necessary.

To overcome this challenge, several solutions were suggested; for instance, the researchers in [80] overcame this challenge in one way: by considering the value chain of the software being built and links in the value chain to help improve usability. According to [27], developers and designers can collaborate with stakeholders using a design studio while maintaining high levels of control.

5) **Documentation:** Documentation is critical for understanding and evaluating prior design decisions in UCD [10].

There is a problem in “Flexible Requirement Development through User Objectives in an Agile-UCD Hybrid Approach.”

According to [39], there was confusion with UX deliverables due to improper documentation of requirements.

To overcome this challenge, the authors in [25] stated that many UCD practitioners use websites as a means of documentation, such as Wikis, to document scenarios, stories, and tasks.

6) **Test:** Since users play a critical role in defining automated acceptance tests by understanding the expected behavior of the software, it is difficult, if not impossible, to create acceptance tests that reflect the human understanding of software usability.

According to [12], traditional user testing sessions are difficult due to the tight schedules associated with Agile environments.

In [18], the authors addressed this challenge by conducting usability testing with the participation of human users. They included usability testing as a separate part of acceptance testing to improve usability and have them in contact with the development team.

5. CONCLUSION

The Agile development process and UX/UCD have become common concepts in software engineering. In this paper, we review the literature on the challenges associated with the integration of Agile and UX/UCD. The studies were reviewed between the years 2000 and 2021, equal to 70, and were gathered from five different research databases: IEEE Xplore Digital Library, ACM Digital Library, Scopus, The Web of Science, and Science Direct. The paper begins with a brief introduction to Agile and UX/UCD. Following that, the research methodology is described, including research questions, search strategy, quality assessment, study selection, and inclusion criteria. Therefore, these study selections are evaluated and organized based on their characteristics. In addition, the limitations of the reviewed studies and future work are investigated and outlined in this article. This review is expected to provide new insights and shed light on recent research in Agile and UX/UCD.

6. REFERENCES

[1] A. Bruun, M. K. Larusdottir, L. Nielsen, P. A. Nielsen, and J. S. Persson, “The role of UX professionals in agile development: A case study from industry,” in 10th Nordic Conference on Human-Computer Interaction, NordiCHI 2018, 2018, pp. 352–363. doi: 10.1145/3240167.3240213.

[2] D. Teka, Y. Dittrich, and M. Kifle, “Adapting Lightweight User-Centered Design with the Scrum-Based Development Process,” in *2018 IEEE/ACM Symposium on Software Engineering in Africa (SEiA)*, 2018, pp. 35–42.

[3] A. Ananjeva, J. S. Persson, and A. Bruun, “Integrating UX work with agile development through user stories: An action research study in a small software company,” *J. Syst. Softw.*, vol. 170, 2020, doi: 10.1016/j.jss.2020.110785.

[4] “Principles behind the Agile Manifesto.” http://agilemanifesto.org/principles.html (accessed Feb. 12, 2022).

[5] S. M. Butt, W. F. W. Ahmad, and L. Rahim, “Handling tradeoffs between agile and usability methods,” in *2014 International Conference on Computer and Information Sciences (ICCOINS)*, 2014, pp. 1–6. doi: 10.1109/ICCOINS.2014.6868450.

[6] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, “Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement,” *BMJ*, vol. 339, no. 7716, pp. 332–336, Jul. 2009, doi: 10.1136/BMJ.B2535.

[7] “Intelligent Diagramming | Lucidchart.” https://www.lucidchart.com (accessed Dec. 28, 2021).

[8] “nails project | Network Analysis Interface for Literature Studies.” https://nailsproject.net/ (accessed Dec. 31, 2021).

[9] “Rayyan – Intelligent Systematic Review.” https://www.rayyan.ai/ (accessed Dec. 31, 2021).

[10] D. Sy, “Adapting Usability Investigations for Agile User-Centered Design,” *J. Usability Stud.*, vol. 2, no. 3, pp. 112–132, May 2007.

[11] J. M. Ungar and J. A. White, “Agile user centered design: Enter the design studio - A case study,” in *28th Annual CHI Conference on Human Factors in Computing Systems*, 2008, pp. 2167–2177. doi: 10.1145/1358628.1358650.

[12] J. M. I. de Carvalho, T. S. da Silva, and M. S. Silveira, “Agile and UCD integration based on pre-development usability evaluations: An experience report,” *18th International Conference on Human-Computer Interaction, HCI International 2016*, vol. 9731. Springer Verlag, ICT-UNIFESP, Universidade Federal de São Paulo, São Paulo, Brazil, pp. 586–597, 2016. doi: 10.1007/978-3-319-39510-4_54.

[13] S. Adikari, C. McDonald, and J. Campbell, “Agile user experience design: A design science enquiry,” 2013. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84923872777&partnerID=40&md5=3f96af86dee0ecd2f8

5a459dbbd2ec31

[14] Z. Husain *et al.*, "Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application," *HCI AND USABILITY FOR EDUCATION AND WORK, PROCEEDINGS*, vol. 5298, no. 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian-Computer-Society. Graz Univ Technol, Inst Software Technol, Graz, Austria, pp. 313-+, 2008.

[15] A. de Oliveira Sousa and N. M. C. Valentim, "Prototyping usability and user experience: A simple technique to agile teams," 2019. doi: 10.1145/3364641.3364667.

[16] C. S. González *et al.*, "Inclusive educational software design with agile approach," in *1st International Conference on Technological Ecosystem for Enhancing Multiculturality, TEEM 2013*, 2013, no. 1st International Conference on Technological Ecosystem for Enhancing Multiculturality (TEEM), pp. 149–155. doi: 10.1145/2536536.2536559.

[17] T. D. Hellmann, A. Hosseini-Khayat, and F. Maurer, "Supporting Test-Driven Development of Graphical User Interfaces Using Agile Interaction Design," in *2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, 2010, pp. 444–447. doi: 10.1109/ICSTW.2010.35.

[18] T. S. d. Silva, M. S. Silveira, and F. Maurer, "Usability Evaluation Practices within Agile Development," in *2015 48th Hawaii International Conference on System Sciences*, 2015, pp. 5133–5142. doi: 10.1109/HICSS.2015.607.

[19] C. S. González-González, P. Toledo-Delgado, and V. Muñoz-Cruz, "Agile human centered methodologies to develop educational software ," *DYNA*, vol. 82, no. 193, pp. 187–194, 2015, doi: 10.15446/dyna.v82n193.53495.

[20] D. Teka, Y. Dittrich, and M. Kifle, "Usability Challenges in an Ethiopian Software Development Organization," in *2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE)*, 2016, pp. 114–120. doi: 10.1109/CHASE.2016.031.

[21] W. Isa, A. M. Lokman, N. F. Taharim, N. D. Wahid, and IEEE, "Engineering M-Learning Using Agile User-Centered Design," *2014 EIGHTH INTERNATIONAL CONFERENCE ON NEXT GENERATION MOBILE APPS, SERVICES AND TECHNOLOGIES (NGMAST)*, no. 8th International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST). Univ Teknol MARA UiTM, Fac Comp & Math Sci, Shah Alam, Selangor, Malaysia NR - 6 PU - IEEE PI - NEW YORK PA - 345 E 47TH ST, NEW YORK, NY 10017 USA, pp. 60–65, 2014. doi: 10.1109/NGMAST.2014.46.

[22] L. Schwartz, A. Vagner, S. Kubicki, and T. Altenburger, "Feedback on the Definition and Design of Innovative Mobile Services," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, 2011, pp. 525–528. doi: 10.1145/2037373.2037453.

[23] K. Kuusinen and T. Mikkonen, "Designing User Experience for Mobile Apps: Long-Term Product Owner Perspective," in *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, 2013, vol. 1, pp. 535–540. doi: 10.1109/APSEC.2013.77.

[24] R. I. Rokhmawati, A. H. Brata, and K. L. Liana, "Perspective-based inspection to improve user experience aspects in SCRUM website's development," in *5th International Conference on Sustainable Information Engineering and Technology, SIET 2020*, 2020, pp. 148–152. doi: 10.1145/3427423.3427459.

[25] H. Obendorf and M. Finck, "Scenario-based usability engineering techniques in agile development processes," in *28th Annual CHI Conference on Human Factors in Computing Systems*, 2008, pp. 2159–2166. doi: 10.1145/1358628.1358649.

[26] M. Singh, "U-SCRUM: An Agile Methodology for Promoting Usability," in *Agile 2008 Conference*, 2008, pp. 555–560. doi: 10.1109/Agile.2008.33.

[27] J. Ungar, "The Design Studio: Interface Design for Agile Teams," in *Agile 2008 Conference*, 2008, pp. 519–524. doi: 10.1109/Agile.2008.51.

[28] V. J. Zhong and T. Schmiedel, "A user-centered agile approach to the development of a real-world social robot application for reception areas," in *2021 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2021*, 2021, pp. 76–80. doi: 10.1145/3434074.3447132.

[29] C. Cuadrat Seix, M. S. Veloso, and J. J. R. Soler, "Towards the Validation of a Method for Quantitative Mobile Usability Testing Based on Desktop Eyetracking," 2012. doi: 10.1145/2379636.2379684.

[30] Z. H. Sain *et al.*, "User interface design for a mobile multimedia application: An iterative approach," in *1st International Conference on Advances in Computer-Human Interaction, ACHI 2008*, 2008, pp. 189–194. doi: 10.1109/ACHI.2008.24.

[31] L. A. Liikkanen, H. Kilpiö, L. Svan, and M. Hiltunen, "Lean UX - The next generation of user-centered Agile development?," in *8th Nordic Conference on Human-Computer Interaction, NordiCHI 2014*, 2014, pp. 1095–1100. doi: 10.1145/2639189.2670285.

[32] E.-M. Schon, J. Thomaschewski, M. J. Escalona, E. Schön, J. Thomaschewski, and M. J. Escalona, "Lean User Research for Agile Organizations," *IEEE Access*, vol. 8, pp. 129763–129773, 2020, doi: 10.1109/ACCESS.2020.3009101.

[33] M. Isomursu, A. Sirotkin, P. Voltti, and M. Halonen, "User Experience Design Goes Agile in Lean Transformation -- A Case Study," in *2012 Agile Conference*, 2012, pp. 1–10. doi: 10.1109/Agile.2012.10.

[34] S. Prior, A. Waller, R. Black, and T. Kroll, "Use of an agile bridge in the development of assistive technology," in *31st Annual CHI Conference on Human Factors in Computing Systems: Changing Perspectives, CHI 2013*, 2013, pp. 1579–1588. doi: 10.1145/2470654.2466210.

[35] C. Felker, R. Slamova, and J. Davis, "Integrating UX with scrum in an undergraduate software development project," in *43rd ACM Technical Symposium on Computer Science Education, SIGCSE'12*, 2012, pp. 301–306. doi: 10.1145/2157136.2157226.

[36] J. Sriarunrasmee and C. Anutariya, "The Development of

One Stop Service Online System based on User Experience Design and AGILE Method," in *11th International Conference on E-Education, E-Business, E-Management, and E-Learning, IC4E 2020*, 2020, pp. 64–69. doi: 10.1145/3377571.3377612.

[37] T. Krohn, M. C. Kindsmüller, and M. Herczeg, "User-centered design meets feature-driven development: An integrating approach for developing the web application myPIM," *1st International Conference on Human Centered Design, HCD 2009. Held as Part of HCI International 2009*, vol. 5619 LNCS. Itemis AG, Schauenburger Str. 116, Kiel 24118, Germany, pp. 739–748, 2009. doi: 10.1007/978-3-642-02806-9_86.

[38] C. Peraire, "Dual-Track Agile in Software Engineering Education," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2019, no. 41st IEEE/ACM International Conference on Software Engineering-Software Engineering Education and Training (ICSE-SEET), pp. 38–49. doi: 10.1109/ICSE-SEET.2019.00013.

[39] M. Budwig, S. Jeong, and K. Kelkar, "When user experience met agile: A case study," in *27th International Conference Extended Abstracts on Human Factors in Computing Systems, CHI 2009*, 2009, pp. 3075–3083. doi: 10.1145/1520340.1520434.

[40] S. Kikitamara and A. A. Noviyanti, "A Conceptual Model of User Experience in Scrum Practice," in *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 581–586. doi: 10.1109/ICITEED.2018.8534905.

[41] L. Miller, "Case study of customer input for a successful product," in *Agile Development Conference (ADC'05)*, 2005, pp. 225–234. doi: 10.1109/ADC.2005.16.

[42] J. Ferreira, "Agile development and UX design: Towards understanding work cultures to support integration," *International Workshops on Advanced Information Systems Engineering Workshops, CAiSE 2012*, vol. 112 LNBIP. Springer Verlag, University of Calgary, Calgary, AB T2N 1N4, Canada, pp. 608–615, 2012. doi: 10.1007/978-3-642-31069-0_51.

[43] N. Pillay and J. Wing, "Agile UX: Integrating good UX development practices in Agile," in *2019 Conference on Information Communications Technology and Society (ICTAS)*, 2019, pp. 1–6. doi: 10.1109/ICTAS.2019.8703607.

[44] D. I. Sensuse, D. Satria, A. A. Pratama, I. A. Wulandari, M. Mishbah, and H. Noprisson, "Integrating UCD into Scrumban for better and faster usability design," in *2017 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2017, pp. 297–302. doi: 10.1109/ICITSI.2017.8267960.

[45] B. Losada, M. Urretavizcaya, J.-M. López-Gil, and I. Fernández-Castro, "Combining InterMod Agile Methodology with Usability Engineering in a Mobile Application Development," 2012. doi: 10.1145/2379636.2379674.

[46] B. Losada, M. Urretavizcaya, and I. Fernández-Castro, "A guide to agile development of interactive software with a 'user Objectives'-driven methodology," *Sci. Comput. Program.*, vol. 78, no. 11, pp. 2268–2281, 2013, doi: 10.1016/j.scico.2012.07.022.

[47] P. Sfetsos, L. Angelis, I. Stamelos, and P. Raptis, "Integrating user-centered design practices into agile Web development: A case study," in *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 2016, pp. 1–6. doi: 10.1109/IISA.2016.7785424.

[48] B. Losada, "Flexible requirement development through user objectives in an Agile-UCD hybrid approach," 2018. doi: 10.1145/3233824.3233865.

[49] M. Seyam, "Enhancing usability through agility: Pair programming for a practice-oriented integration approach," in *2015 International Conference on Collaboration Technologies and Systems (CTS)*, 2015, pp. 460–463. doi: 10.1109/CTS.2015.7210467.

[50] K. A. Abdelouhab, D. Idoughi, and C. Kolski, "Agile &amp; user centric SOA based service design framework applied in disaster management," in *2014 1st International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, 2014, pp. 1–8. doi: 10.1109/ICT-DM.2014.6917792.

[51] S. Bordin and A. De Angeli, "Focal points for a more user-centred agile development," *17th International Conference on Agile Processes in Software Engineering and Extreme Programming, XP 2016*, vol. 251. Springer Verlag, Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 9, Trento, 38123, Italy, pp. 3–15, 2016. doi: 10.1007/978-3-319-33515-5_1.

[52] S. R. Humayoun, Y. Dubinsky, and T. Catarci, "A three-fold integration framework to incorporate user-centered design into agile software development," *2nd International Conference on Human Centered Design, HCD 2011, Held as Part of HCI International 2011*, vol. 6776 LNCS. Dipartimento di Informatica e Sistemistica A. Ruberti, SAPIENZA, Università di Roma, Via Ariosto - 25, Roma 00185, Italy, pp. 55–64, 2011. doi: 10.1007/978-3-642-21753-1_7.

[53] P. L. M. Navarro, G. M. Pérez, and D. S. Ruiz, "A Script-Based Prototyping Framework to Boost Agile-UX Developments," *J. Comput. Sci. Technol.*, vol. 31, no. 6, pp. 1246–1261, 2016, doi: 10.1007/s11390-016-1695-6.

[54] A. L. Peres and S. L. Meira, "Towards a framework that promotes integration between the UX design and SCRUM, Aligned to CMMI," in *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*, 2015, pp. 1–4. doi: 10.1109/CISTI.2015.7170443.

[55] C. S. A. Peixoto, "Human-computer interface expert system for agile methods," in *ITI 2009 31st International Conference on Information Technology Interfaces, ITI 2009*, 2009, pp. 311–316. doi: 10.1109/ITI.2009.5196100.

[56] S. Anwar, Y. H. Motla, Y. Siddiq, S. Asghar, M. S. Hassan, and Z. I. Khan, "User-centered design practices in scrum development process: A distinctive advantage?," in *17th IEEE International Multi Topic Conference 2014*, 2014, pp. 161–166. doi: 10.1109/INMIC.2014.7097330.

[57] Y. Xiong and A. Wang, "A new combined method for

UCD and software development and case study," in *The 2nd International Conference on Information Science and Engineering*, 2010, pp. 1–4. doi: 10.1109/ICISE.2010.5690032.

[58] N. K. Al Ghanmi and N. S. M. Jamail, "Integrating scrum development process with ux design flow," *Bull. Electr. Eng. Informatics*, vol. 9, no. 6, pp. 2630–2636, 2020, doi: 10.11591/eei.v9i6.2484.

[59] C. Moralles *et al.*, "On the Mapping of Underlying Concepts of a Combined Use of Lean and User-Centered Design with Agile Development: The Case Study of the Transformation Process of an IT Company," *10th Brazilian Workshop on Agile Methods, WBMA 2019*, vol. 1106 CCIS. Springer, MunDDoS Research Group – PPGCC – School of Technology, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brazil, pp. 25–40, 2019. doi: 10.1007/978-3-030-36701-5_3.

[60] M. Zorzetti *et al.*, "An Empirical-informed Work Process Model for a Combined Approach of Agile, User-Centered Design, and Lean Startup," 2020. doi: 10.1145/3439961.3439967.

[61] V. Paelke and K. Nebe, "Integrating agile methods for mixed reality design space exploration," in *7th ACM Conference on Designing Interactive Systems - DIS 2008*, 2008, pp. 240–249. doi: 10.1145/1394445.1394471.

[62] D. Salah, R. Paige, and P. Cairns, "A maturity model for integrating agile processes and user centred design," *16th International Conference on Software Process Improvement and Capability Determination, SPICE 2016*, vol. 609. Springer Verlag, Sadat Academy for Management Science, Cairo, Egypt, pp. 109–122, 2016. doi: 10.1007/978-3-319-38980-6_9.

[63] E. Kropp and K. Koischwitz, "User-centered-design in agile RE through an On-site User Experience Consultant," in *2014 IEEE 2nd International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*, 2014, pp. 9–12. doi: 10.1109/UsARE.2014.6890994.

[64] E. Kropp and K. Koischwitz, "Experiences with user-centered design and agile requirements engineering in fixed-price projects," *1st International Workshop on Usability- and Accessibility-Focused Requirements Engineering, UsARE 2012, Held in Conjunction with 34th International Conference on Software Engineering, ICSE 2012, 2nd International Workshop, Usability- and Accessibility-Fo*, vol. 9312 LNCS. Springer Verlag, Akquinet, Berlin, Germany, pp. 47–61, 2016. doi: 10.1007/978-3-319-45916-5_4.

[65] H. Williams and A. Ferguson, "The UCD Perspective: Before and After Agile," in *Agile 2007 (AGILE 2007)*, 2007, pp. 285–290. doi: 10.1109/AGILE.2007.61.

[66] J. Ferreira, J. Noble, and R. Biddle, "Agile Development Iterations and UI Design," in *Agile 2007 (AGILE 2007)*, 2007, pp. 50–58. doi: 10.1109/AGILE.2007.8.

[67] M. Najafi and L. Toyoshiba, "Two Case Studies of User Experience Design and Agile Development," in *Agile 2008 Conference*, 2008, pp. 531–536. doi: 10.1109/Agile.2008.67.

[68] D. Teka, Y. Dittrich, and M. Kifle, "Integrating discount usability in scrum development process in Ethiopia," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 2017, pp. 1–8. doi: 10.1109/ICCNI.2017.8123811.

[69] P. Hodgetts, "Experiences integrating sophisticated user experience design practices into agile processes," in *Agile Development Conference (ADC'05)*, 2005, pp. 235–242. doi: 10.1109/ADC.2005.24.

[70] S. Lee, I. Ko, S. Kang, and D. Lee, "A Usability-Pattern-Based Requirements-Analysis Method to Bridge the Gap between User Tasks and Application Features," in *2010 IEEE 34th Annual Computer Software and Applications Conference*, 2010, pp. 317–326. doi: 10.1109/COMPSAC.2010.39.

[71] J. M. Brown, G. Lindgaard, and R. Biddle, "Collaborative Events and Shared Artefacts: Agile Interaction Designers and Developers Working Toward Common Aims," in *2011 Agile Conference*, 2011, pp. 87–96. doi: 10.1109/AGILE.2011.45.

[72] I. Klooster, M. L. Noordzij, and S. M. Kelders, "Exploring how professionals within agile health care informatics perceive visualizations of log file analyses: Observational study followed by a focus group interview," *JMIR Hum. Factors*, vol. 7, no. 1, 2020, doi: 10.2196/14424.

[73] K. Kuusinen and K. Väänänen-Vainio-mattila, "How to make agile UX work more efficient: Management and sales perspectives," in *7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, NordiCHI 2012*, 2012, pp. 139–148. doi: 10.1145/2399016.2399037.

[74] J. Ferreira, H. Sharp, and H. Robinson, "User experience design and agile development: Managing cooperation through articulation work," *Softw. - Pract. Exp.*, vol. 41, no. 9, pp. 963–974, 2011, doi: 10.1002/spe.1012.

[75] T. S. Da Silva, M. S. Silveira, C. De O. Melo, and L. C. Parzianello, "Understanding the UX designer's role within agile teams," *2nd International Conference on Design, User Experience, and Usability: Design Philosophy, Methods, and Tools, DUXU 2013, Held as Part of 15th International Conference on Human-Computer Interaction, HCI International 2013*, vol. 8012 LNCS, no. PART 1. ICMC, USP, Universidade de São Paulo Campus de São Carlos, Brazil, pp. 599–609, 2013. doi: 10.1007/978-3-642-39229-0_64.

[76] J. Brown, G. Lindgaard, and R. Biddle, "Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts," in *Agile 2008 Conference*, 2008, pp. 39–50. doi: 10.1109/Agile.2008.54.

[77] T. Øvad *et al.*, "Teaching Software Developers to Perform UX Tasks," in *27th Australian Computer-Human Interaction Conference, OzCHI 2015*, 2015, pp. 397–406. doi: 10.1145/2838739.2838764.

[78] J. Ferreira, H. Sharp, and H. Robinson, "Agile Development and User Experience Design Integration as an Ongoing Achievement in Practice," in *2012 Agile Conference*, 2012, pp. 11–20. doi: 10.1109/Agile.2012.33.

[79] O. Almughram and S. Alyahya, "Coordination support for integrating user centered design in distributed agile

projects,” in *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2017, pp. 229–238. doi: 10.1109/SERA.2017.7965732.

[80] D. Kane, “Finding a place for discount usability engineering in agile development: throwing down the gauntlet,” in *Proceedings of the Agile Development Conference, 2003. ADC 2003*, 2003, pp. 40–46. doi: 10.1109/ADC.2003.1231451.

[81] T. S. Da Silva, M. S. Silveira, and F. Maurer, “Usability evaluation practices within agile development,” in 48th Annual Hawaii International Conference on System Sciences, HICSS 2015, 2015, vol. 2015-March, pp. 5133–5142, doi: 10.1109/HICSS.2015.607.