A Systematic Mapping Study on Agile UCD in Software Engineering Research

Muna Al-Razgan, Nuha Alrajhi, Samar Alqahtani, Reham Althunayan, Manal Alshehri, Nafla Alrumayan and Amal Alblwai
Department of Software Engineering, College of Computer and Information Sciences,
King Saud University, Riyadh, 11345, Saudi Arabia

ABSTRACT
This study provides a comprehensive overview of how Agile and User-Centred Design (UCD) methodologies are integrated within software engineering research. By methodically analysing fifty research papers. We investigated the use of various Human-Computer Interaction (HCI) techniques and research methods. Our review reveals a diverse application of techniques such as "Hybrid" and "Collaborative Design," highlighting their practical implementations, predominantly illustrated through case studies. The research output is grouped into categories including Experience, Solution, Evaluation, and Validation papers, reflecting an active community eager to share insights and innovate continuously. Key findings indicate that the synergy of Agile and UCD methodologies significantly boosts user engagement and satisfaction, enhances usability, and refines development processes. These benefits demonstrate the crucial impact of these methodologies on enhancing the efficiency of development practices and improving the quality of software products. The insights from this study highlight existing approaches and inform future research directions, emphasizing the importance of evolving Agile and UCD methodologies to meet changing user needs and software development trends.

Keywords
Agile Software Development, User-Centred Design, Human-Computer Interaction, Systematic Mapping, Empirical Software Engineering.

1. INTRODUCTION
The integration of Agile methodologies and User-Centred Design (UCD) within software engineering has garnered significant attention due to their potential to enhance both the efficiency of development processes and the effectiveness of the resulting software products. Agile methodologies are lauded for their adaptability and iterative nature, which facilitates rapid responses to changing project requirements and emphasizes continuous improvement and customer satisfaction [1]. This approach is rooted in the principles of the Agile Manifesto, which emphasizes continuous delivery of valuable software [2]. Conversely, User-Centred Design focuses on optimizing user interfaces and ensuring product usability by involving users throughout the design process, thus guaranteeing that the products are not only functional but also user-friendly [3].

Despite the individual strengths of each methodology, integrating these two approaches presents a unique set of challenges and opportunities. The dynamic and often fast-paced environment of Agile can conflict with the methodical and sometimes slower processes of UCD, which requires thorough user research and testing. However, when successfully integrated, these methodologies can complement each other, leading to software products that are efficiently developed and deeply aligned with user expectations.

This paper presents a systematic mapping study on Agile UCD, exploring how these methodologies have been integrated within the realm of software engineering research. It aims to address gaps in the literature and identify successful practices and persistent challenges in combining Agile and UCD. Building on foundational works such as Silva da Silva et al. [4], which examined the intersection of Agile methodologies and UCD within the Human-Computer Interaction (HCI) community, this study extends these insights to broader software engineering practices.

The following sections of this paper will delve into the specific characteristics of Agile methodologies and UCD, explore the reasons for their integration, and detail the methodological framework employed in this study. This investigation seeks to bridge the gap between agile flexibility and User-Centred thoroughness, offering insights into how these methodologies can be synergistically applied to improve both the process and products of software development.

2. BACKGROUND
In the dynamic field of software development, integrating various methodologies to enhance product quality and user satisfaction is always a goal. Among these methodologies, Agile and UCD stand out as influential frameworks, each offering unique strengths to the development process. Agile is known for its rapid delivery and flexibility, while UCD is valued for its deep focus on user needs and usability. This introduction explores the distinct characteristics of Agile and UCD, their integration into a cohesive development strategy, and the challenges that such an integration presents. We will cover these topics in the following four main sections.

A. Agile and User-Centred Design (UCD)
In the rapid realm of software engineering, Agile methodologies are known for being fast and flexible. These methods are part of a software development philosophy that adheres to the principles outlined in the Agile Manifesto, which prioritizes customer satisfaction through delivering valuable software early and continuously [2]. Agile is characterized by its focus on continuous improvement, rapid delivery, and responsiveness to changing customer needs. It differentiates itself from traditional models by expecting the evolution of project requirements and solutions throughout the development process and by breaking down tasks into small, manageable increments that allow for iterative improvements [5].

On the other hand, UCD is widely recognized as a key framework for developing user interfaces. It relies heavily on the active involvement of users to enhance the understanding of user and task requirements. UCD is an iterative process that includes repeated cycles of design and evaluation to ensure the end product truly meets the needs

of its users [3][1].

B. The Integration of UCD Into Agile Processes

Despite the widespread adoption of Agile methodologies over the past two decades, certain limitations have become apparent, particularly regarding user involvement. The integration of UCD with Agile—forming a hybrid development methodology—is increasingly seen as an interesting solution for these problems, drawing attention from both researchers and professionals [6]. Some critics believe that Agile by itself sometimes falls short in addressing business-level issues and often provides insufficient customer involvement [7]. Merging Agile with other methods like Lean Startup and UCD has been suggested to fix these issues. This combination seems promising as it increases involvement from stakeholders and allows for quick experiments, among other advantages [8].

At first glance, UCD might seem to conflict with Agile; UCD often requires a lot of research before starting the development, while Agile wants to start delivering quickly in short cycles [9]. However, both methodologies prioritize the end-user and customer. According to the Interaction Design Foundation, UCD is fundamentally an iterative process that emphasizes user needs at every stage of the design process [10]. Therefore, bringing Agile and UCD together aims to use the best parts of both: Agile's speed and UCD's focus on users. Achieving a good balance between these methodologies is crucial to ensure that Agile's flexibility and UCD's comprehensive user engagement effectively complement each other.

C. The Need for Agile/UCD Integration.

Agile methodologies and UCD both aim to produce high-quality software but approach this goal from different angles. Agile focuses on creating functional code and features quickly through iterative cycles, prioritizing customer satisfaction. In contrast, UCD emphasizes enhancing usability and user experience, incorporating detailed user feedback at each stage [11],[12]. This distinction highlights that while Agile targets quick delivery based on customer feedback, UCD delves into extensive user research to refine product design.

The integration of UCD with Agile methodologies addresses the limitations of each approach by combining their strengths. This combination allows Agile to keep its quick development speed while incorporating UCD's detailed assessment of user needs. As a result, the product becomes easier to use and its overall quality improves. This approach also integrates structured user feedback into Agile's rapid cycles, promoting a more comprehensive development strategy that considers the needs of both customers and users. The benefits of this hybrid method include improved communication and collaboration between the development and design teams, which lead to more efficient development processes and higher-quality results [13].

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

D. Challenges in Agile/UCD Integration

Integrating Agile methodologies with UCD presents several challenges due to differences in their approaches and roles. One significant issue is that Agile does not specify a distinct role for designers, complicating the integration of UCD. There is an absence of clear guidelines on how UX designer roles should be defined and integrated within Agile development teams [14]. Additionally, aligning the iterative cycles of design and development activities, which involves coordination of timing and scheduling, poses significant challenges [12].

Another challenge arises from how progress is defined and measured in Agile methodologies—primarily through the creation of working software. This focus can make it difficult to prioritize UX-related decisions that might require more attention to user experience rather than immediate functionality [11],[15].

Moreover, Agile and UCD differ significantly in their approaches to resource allocation, particularly in terms of upfront work. Agile prefers extensive initial work to stay adaptable to changing requirements across development iterations. In contrast, UCD emphasizes in-depth upfront user research and analysis to thoroughly understand user needs before development begins. This basic difference in approach makes it harder to integrate Agile and UCD effectively [14].

3. RELATED WORK

Several pivotal studies have conducted systematic literature reviews (SLR) and mapping studies on Agile UCD publications. In early studies, Silva da Silva, Tiago, et al. [4] delve into the intersection of Agile methodologies and UCD within the HCI community. Their systematic mapping study methodology reviews contributions from significant HCI and Agile conferences, identifying prevalent trends, methods, and techniques demonstrating a robust integration of UCD in Agile environments.

In 2019, a comprehensive systematic mapping by [16] Ogunyemi et al. assessed the existing methodologies employed in HCI during software development. This study, which provides a broad overview of the methodologies and techniques employed in HCI practice and their effectiveness, has revealed various methods and techniques applied in software development. Significantly, it has emphasized the growing acknowledgment of HCI methods and their incorporation into diverse software engineering processes. The study's findings contribute substantially to comprehending the HCI practice environment and provide valuable insights for future research and practice. Similarly, in an insightful article [17], Ferreira and Canedo present an SLR focused on the integration of Design Sprints (DS) with project-based learning (PBL) in software engineering education. The SLR, which aimed to explore the effectiveness of incorporating DS methodologies to enhance the authenticity and user experience considerations in software development projects students undertake, discussed various aspects of the application, including the challenges of adapting DS to fit within the academic schedules and the impact on student engagement and learning outcomes.

At the same time, Sousa and Valentim [18] integrated systematic literature mapping (SLM) and empirical

methods, specifically through pilot studies and feasibility studies with students and planning professionals, to refine and validate the effectiveness of User Experience and Usability Guidelines for Agile Project (UXUG-AP) technique in practical settings. The research addresses the challenge of efficiently incorporating usability and UX in agile projects, providing agile teams with practical, cost-effective, and flexible usability and UX guidelines.

The SLR in [19] identifies and evaluates UCD techniques specifically for mobile applications. The authors meticulously followed established SLR guidelines to assess the efficacy of UCD methods in enhancing mobile app usability and interface design. The study's findings consolidate effective practices and highlight gaps within the existing research, providing valuable insights for developers and designers focused on mobile user experience.

In 2022, the study [20] by Hinderks et al. explored the integration of User Experience (UX) approaches into Agile software development frameworks. Through a systematic literature review methodology, the authors unearth strategies that facilitate incorporating UX practices into Agile processes, revealing a diverse range of approaches. These include upfront UCD, enhanced communication between development and UX teams, and integrating specific UX methods like prototyping and user testing into Agile cycles. The study's findings underscore the absence of a standardized approach to managing UX in Agile environments, underscoring the need for further research to define and optimize UX integration in Agile methodologies. Leinonen and Roto [21] explore the critical knowledge transfer phase between service design and UX design in digital Service Creation Projects. This study employs a systematic literature review to understand and enhance the flow of information across these stages, emphasizing the challenges and solutions associated with the handover process. Furthermore, the authors identify and discuss strategies to improve communication and effectively use boundary objects in this context. They propose frameworks for managing knowledge transfer and outline best practices to ensure successful information flow throughout service creation, addressing an underexplored yet crucial area of project management and design interaction.

Alabood et al. [22] 2023 explored the application of Design Critique (DC) within the context of HCI and UX design. The research systematically reviews the literature through thematic analysis to understand how DC can be effectively incorporated into the design process, especially in agile environments. The authors highlight the potential of DC for enhancing usability and system development progress by integrating iterative feedback mechanisms typical of agile methodologies.

Our systematic mapping provides a state-of-the-art Agile UCD in software engineering research and draws inspiration from the Silva da Silva et al.' study [4]. Furthermore, we investigate and try to collect HCI techniques based on the study [16], which will help inform researchers and practitioners of the current state of the art and specify future areas for research and practice development in software engineering.

4. RESEARCH METHODOLOGY

A systematic mapping study's primary objective is to give an overview of the current research field, identify the types and quantities of research that have been done up to date, and identify the results that have been obtained in the context of UCD and HCI techniques in software engineering research. To identify trends, it is common to map publication frequencies over time. Finding the forums where relevant research has been published can also be a secondary objective [23].
This systematic mapping review was conducted to answer the research questions listed below. A comprehensive review was carried out using a variety of studies, in which the research questions were addressed, and the most relevant studies were consolidated into one classification.
This study addresses four key research questions: RQ1 explores the HCI techniques used to integrate Agile and UCD, seeking to categorize methods that harmonize Agile's flexibility with UCD's user-focused principles; RQ2 examines the research methods employed in Agile UCD studies, aiming to classify methodologies such as controlled experiments and ethnographies; RQ3 investigates the types of papers published on Agile UCD, categorizing them into various scholarly forms like research articles and theoretical frameworks; and RQ4 evaluates the proposed benefits of these publications, such as improved communication and usability.

E. Conducted Search

The study must be a full-length paper written in English, published between 2019 and 2023, relevant to the main topic, and within the area of Software Engineering.

We decided to conduct the search in the last five years by manually searching through the databases. We aimed to use different data sources to get relevant studies as possible. We used several digital libraries that are well known in the software engineering field. The search process was manually conducted by searching through databases via specific protocol depending on the database advanced search properties. The selected databases are as follows: IEEE Xplore, ACM Library, Springer Link, ISI Web of Science, and ScienceDirect.

We aimed to search using the following keyword protocol: (("Document Title": “User-Centred Design OR "Document Title": “User Experience" OR "Document Title": UX OR "Document Title":UCD OR "Document Title":"User Centred Design") AND ("Document Title": Agile OR "Document Title": Kanban OR "Document Title": "Extreme Programming" OR "Document Title": Lean OR "Document Title": Scrum)) OR (("Abstract": "User-Centred Design" OR "Abstract": UCD OR "Abstract": "User Centred Design") AND ("Abstract": Agile OR "Abstract": Kanban OR "Abstract": "Extreme Programming" OR "Abstract": Lean OR "Abstract": Scrum)).

The search was conducted using both title and abstract. The total number of papers in the main databases between the years 2019 and 2023 was 328 studies.

F. Screening Papers

After collecting this initial set of 328 papers, we exported the citations from each of the previously mentioned databases. The numbers are illustrated in Figure 1 below. The highest number of papers was found in ISI Web of Science with 133 studies, followed by, ACM Library with 78 studies, IEEE Xplore: 44 studies, SpringerLink, with a total of 41 studies, and finally ScienceDirect: 32 studies.

There was a chance of encountering duplicated studies within the search, therefore, the next step was mandatory to eliminate the duplicates. To do so, we used Rayyan QCRI, Rayyan QCRI is a free web and mobile application that helps expedite the initial screening of both abstracts and titles through a semi-automated process while incorporating a high level of usability. Its main benefit is to speed up the most tedious part of the systematic literature review process: selecting studies for inclusion in the review [24]. Therefore, for this step, we exported all the resulting studies to Rayyan to check for duplicates as well as scanning the results to decide which to include. Using Rayyan, a total of 33 duplicate studies were found, resulting in a total of 295 unique studies.

Using Rayyan, as we resulted in 295 studies, we needed more filtration to be conducted between the six researchers regarding the abstract. To minimise risk of bias we turned on the blind mode in Rayyan QCRI. When blinding is on, decisions, labels and notes are not visible to other collaborators. Each researcher took a comprehensive look at each abstract of 295 studies. When turning the blind mode off in Rayyan the results in Figure 2 were found among collaborators. All researchers agreed to exclude 174 studies, 82 studies to be included, 27 conflicted in decision among researchers, 12 studies categorised as “Maybe”.

The researchers then held multiple sessions in which it was decided whether to include or exclude each of the conflicted studies. We agreed that once at least four researchers agree, an immediate decision was made, and the study were either moved to included or excluded. Conversely, there have been studies where fewer than four researchers reached a consensus on any given decision. In these cases, the collaborators had to reread the abstract one more time and provide additional context to persuade the other researchers whether to include or exclude

This filtration was conducted to guarantee paper's relevance to the main subject of this systematic mapping, as well as conceiving general agreement among researchers upon any conflicted paper. After filtering based on the abstracts, 225 papers were agreed to be excluded due to irrelevance. At this stage, 70 studies remained. The second step of filtration eliminated 17 more studies, where all studies were unfortunately inaccessible by the researchers.

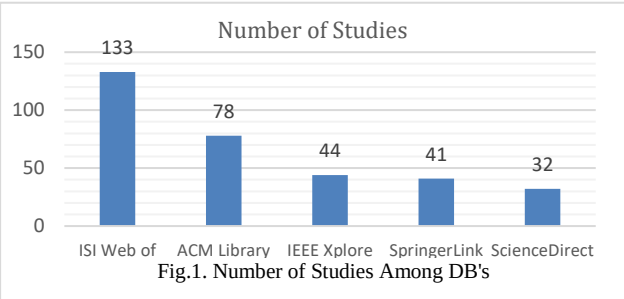


Fig.1. Number of Studies Among DB's

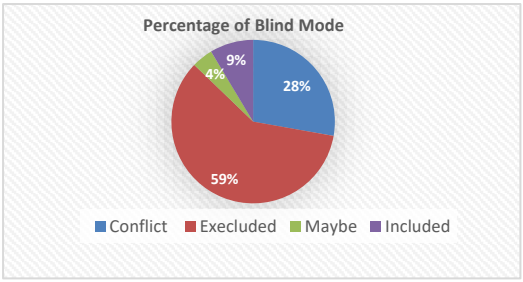


Fig. 2. Results of Blind Mode

After this step, a total of 53 studies were included in this systematic mapping.

G. Analyzing Stage

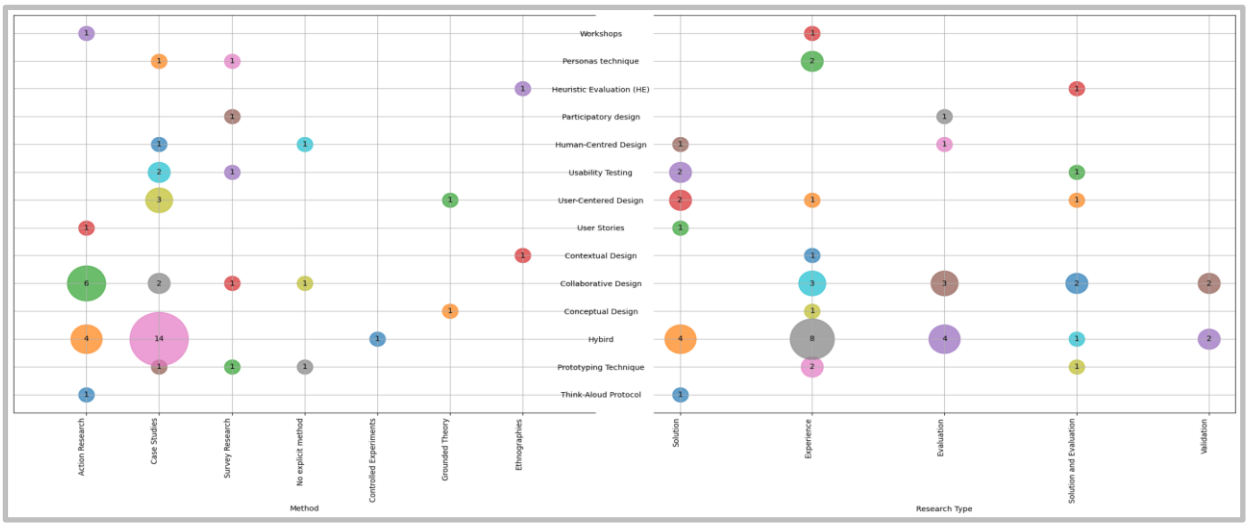
All researchers collected the data. The papers were distributed equally between them, and each researcher read each paper completely to determine its topic, extract HCI technique used, types of papers on Agile UCD, type of research method used, and the benefits in integrating Agile and UX, write a summary about it, and record this information in a common shared Excel spreadsheet. At this stage, the authors also eliminated three additional studies upon reading the full-length study, as they were not directly related to the main subject. We resulted in 50 related papers included in this systematic mapping study.

5. RESULTS

In this section, we present the results derived from the analysis of the 50 primary studies. The systematic map is represented through three-dimensional bubble charts plotted on x-y coordinates to illustrate the intersection and integration of HCI techniques with Agile and UCD methodologies in software development [23]. The map provides a comprehensive overview of the 50 primary studies analysed. The categorization scheme, detailed in Table 3, serves as the basis for our systematic map, which is displayed in Figure 3. We find that only 14 out of the 59 techniques listed in Table 4 are utilized across various research types and have been integrated through diverse methods, highlighting the selective yet impactful use of these techniques in real-world applications.

"Hybrid" and "Case Study" are the most frequently used HCI techniques and research methods in the Agile and UCD field, respectively. The data clearly shows a strong trend toward 'Hybrid' HCI techniques, which appear 19 times across the chart in various research categories, most commonly with case studies and action research methods. This indicates that in the dynamic, iterative environments typical of Agile projects, 'Hybrid' techniques are crucial for integrating User-Centred design principles more effectively, allowing researchers to adapt and improve their approaches in response to evolving project needs and user feedback. Notably, 'Hybrid' techniques also dominate the 'Experience' aspect of HCI, where they appear eight times. This consistent use emphasizes the importance of experience-focused studies in evaluating the practical

Fig. 3. Systematic map – HCI Techniques, Methods, and Research Type



effects of HCI techniques on user interaction and satisfaction in real-world scenarios. Following 'Hybrid', 'Collaborative Design' stands out as another frequently used technique, evident in its broad application across various research types and methods. It is applied three times each within 'Experience' and 'Evaluation' contexts, highlighting its crucial role in fostering cooperative and participatory design processes that are key to effective Agile UCD practices. Additionally, it is employed six times in conjunction with action research.

On the right side of the chart, the distribution of HCI techniques in the 'Experience' category is clearly evident, with an occurrence of 19 papers. Additionally, the 'Experience ' research type is extensively employed in 'Hybrid ' and 'Collaborative Design ' projects. This widespread use emphasizes the critical role that user and developer experiences play in the associated research, highlighting their fundamental contribution to the development of User-Centred design solutions. However, this focus on experiential methodologies leaves empirical research methods less explored. This gap suggests there are significant opportunities to further investigate these approaches to fully understand their impacts in real-world agile settings.

In terms of research methods, 'Case studies' are frequently employed, appearing 24 times with various HCI techniques and 14 times specifically with 'Hybrid' techniques. This widespread use underscores their importance in exploring and validating the application of these methodologies in diverse contexts. However, there is a notable gap in the application of empirical research methods. This suggests a significant area for future investigation, where empirical research could assess the effectiveness and adaptability of various HCI techniques in agile environments.

The Figure 3 shows that certain HCI techniques, like 'Heuristic Evaluation' and 'Persona's technique,' are rarely utilized across different research types. This limited use indicates that these areas are underexplored, suggesting that future research could provide valuable insights, especially in improving design and evaluation processes within Agile UCD projects. This observation highlights a significant gap in the current research, specifically in the integration of HCI techniques with agile methodologies using detailed research methods. There is a noticeable shortage of studies employing Controlled Experiments, Grounded Theory, and Ethnographic methods. This underscores the need for increased research efforts that use these rigorous methodologies to deepen our understanding of HCI in agile environments.

H. Classification scheme

1. HCI Techniques

- Think-Aloud Protocol: This technique involves users verbally expressing their thoughts while interacting with a system, allowing researchers to gain insight into the user's cognitive processes.
- Prototyping Technique: This approach involves creating a preliminary, less complex model of the system or product to explore ideas and validate functionality early in the design process.
- Hybrid Approach: This technique is a mixed method that combines various HCI techniques to leverage the strengths of multiple approaches.
- Conceptual Design: This process focuses on system conceptualization by developing and refining the ideas and principles that underlie the features, functionalities, and overall structure of a system.
- Collaborative Design: This design approach involves multiple stakeholders actively participating in the design process, ensuring that diverse perspectives are considered.

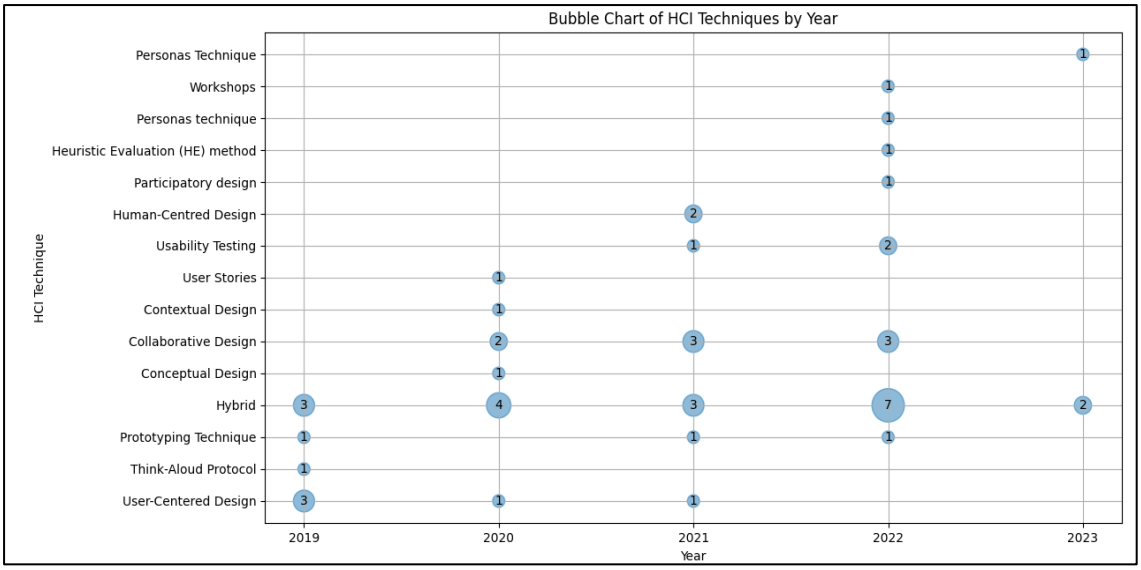


Fig. 4. HCI Technique Among Years

- Contextual Design: This method involves observing design decisions within their natural context, considering the environment, workflows, and goals to ensure the design aligns with real-world use. User Stories: This technique provides features from the viewpoint of the end user, focusing on what users need and expect from the system.
- User-Centred Design (UCD): UCD is a framework that places significant emphasis on usability goals,
- Participatory Design: This inclusive approach actively engages all stakeholders, particularly end-users, in the design process to ensure that the final outcome meets their requirements.
- Heuristic Evaluation Method: This method involves experts using established heuristics or rules of thumb to evaluate the usability of a system.
- Persona Technique: This technique involves developing fictional characters based on user research to represent distinct user archetypes.
- Workshops: These are interactive sessions held to explore, debate, and iterate on various aspects of system design with stakeholders.

2. *Research Types*
   - Solution Research: This type of research focuses on developing new Agile methods or tools that effectively integrate UCD principles to enhance user experience and software usefulness. It often includes providing a proof-of-concept through examples or logical arguments.
   - Validation Research: This research examines the effectiveness of Agile UCD methods or tools in real-world software development environments, aiming to achieve user-centered outcomes. It utilizes rigorous methods such as experiments, simulations, prototyping, or mathematical analyses to demonstrate the practicality and effectiveness of these solutions.
   - Experience Research: This research gathers user characteristics, environment, tasks, and workflow at every stage of the design process.
   - Usability Testing: This process assesses a product or service by testing it with users, observing them as they attempt to complete tasks to identify usability issues.
   - Human-Centred Design: This approach focuses on solving problems by incorporating the human perspective at every step of the problem-solving process.
     insights from software development teams or users regarding their experiences with Agile UCD, highlighting the challenges, successes, and practical effects of its implementation. It emphasizes learning from direct, practical experiences.
   - Evaluation Research: This type of research assesses the impact of Agile UCD practices on software project outcomes, such as user satisfaction, development efficiency, and product quality. It may involve empirical methods like case studies or surveys to understand cause and effect.
   - Solution and Evaluation Research: This research explores the development of new Agile UCD practices while rigorously evaluating their practical effectiveness in improving user engagement and project success.

3. *Research Methods*
   - Grounded Theory (GT): This research methodology allows for the development of theories grounded in systematically gathered and analyzed data. It emphasizes iterative analysis and the development of emergent themes through coding and recoding[25].
   - Case Studies: This empirical method involves analyzing a specific modern phenomenon within its actual context, particularly when the distinctions between the phenomenon and the

environment are not readily apparent. It can be explanatory, exploratory, or descriptive [25].

- Survey Research: This method is a form of inquiry used to facilitate change within groups or communities, where the researcher actively participates and collaborates with practitioners in problem-solving [25].
- Controlled Experiments: This research method involves manipulating independent variables to measure their effect on dependent variables in a controlled setting. It is used to validate hypotheses about the effects of different conditions or treatments in software development practices [25].
- Action Research: This method is a collaborative form of inquiry used to facilitate change within groups or communities, involving the researcher's active participation in problem-solving alongside practitioners [25].
- Ethnographies: This method involves a detailed and systematic study of people and cultures from the perspective of the subject, aimed at understanding software development practices within their natural settings[25].
- The Human-Computer Interaction (HCI) techniques, which are methods and approaches used in designing user-centered systems. Some techniques focus on user research, such as Personas, Interviews, Ethnography, and Contextual Inquiry, which help in understanding user needs, behaviors, and contexts. Methods like Affinity Diagramming, Card Sorting, Storyboarding, and Wireframing aid in organizing ideas and creating visual representations of systems. There are also techniques for evaluating and testing usability, including Expert Review, Heuristic Evaluation, Cognitive Walkthrough, A/B Testing, User Testing, and Remote Usability Testing. Collaborative approaches like Participatory Design, Workshops, Focus Groups, and Co-discovery Usability Testing emphasize involving multiple stakeholders and users in the design process. Furthermore, techniques such as Prototyping, Mock-ups, Sketching, and Design Studio are used to create and refine early versions of a system. Other methods mentioned, like Human-Centred Design, Collaborative Design, Conceptual Design, and frameworks like Lean UX and Garrett's Framework, focus on integrating user-centered principles throughout the design process. Each of these techniques serves a specific purpose, from generating ideas to evaluating user experiences, ensuring that the final product is both functional and user-friendly.

In regards of the HCI techniques used for the integration of agile and UCD, as it is illustrated in Figure 4 below, "Hybrid" HCI techniques had the highest total number of studies over the years, with a total of 19 studies where the authors of these studies used more than one HCI Technique, followed by "Collaborative Design" with 8 studies, and "User-Centred Design" with 5 studies. Conversely, some techniques had fewer, or no studies conducted, such as "Think-Aloud Protocol" with only 1 study [26-28].

Upon the results, the average number of studies conducted per year for each technique reveals insightful patterns in research activity. Techniques like "Hybrid" and "Collaborative Design" demonstrated sustained interest, with averages of approximately 4 and 2 studies per year, respectively, indicating consistent research efforts. Conversely, techniques such as "Think-Aloud Protocol" and "Contextual Design" showed lower averages, suggesting intermittent research activity or specialised application. For instance, "Think-Aloud Protocol" had an average of 1 study per year, while "Contextual Design" had an average of less than 1 study per year. These metrics highlight variations in research attention across different HCI techniques, providing valuable context for understanding the dynamics of research trends and areas of focus within the field. Lastly, as in Figure 5 below, we can examine the distribution of studies across the years. This analysis helps identify trends or spikes in research activity. For instance, "Hybrid" techniques saw a significant increase in studies in 2022 compared to previous years, suggesting a growing interest or emerging trends in that year. Conversely, some techniques may have fluctuating or sporadic research activity across the years.

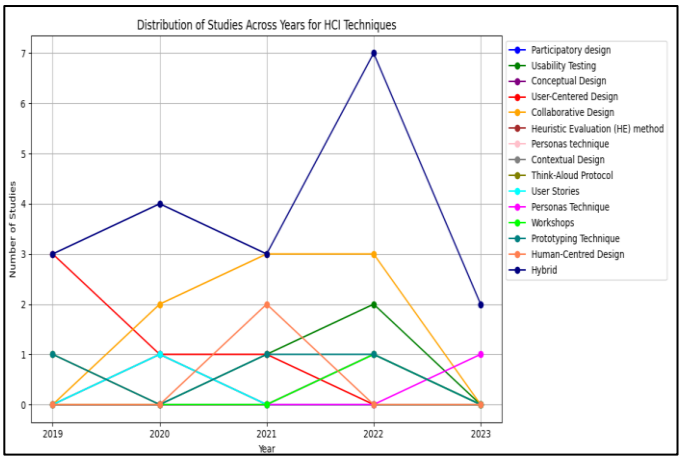


Fig.5. Distribution of Studies Across the Years

The questions provided focus on different aspects of research in Agile User-Centred Design (UCD) studies.

For RQ1, the question investigates the types of research methods used in Agile UCD studies. Silva da Silva et al. identified six methods: Case Studies, Survey Research, Ethnographies, Action Research, Grounded Theory, and Controlled Experiments. The analysis shows that from 2019 to 2023, Case Studies were the most frequently used

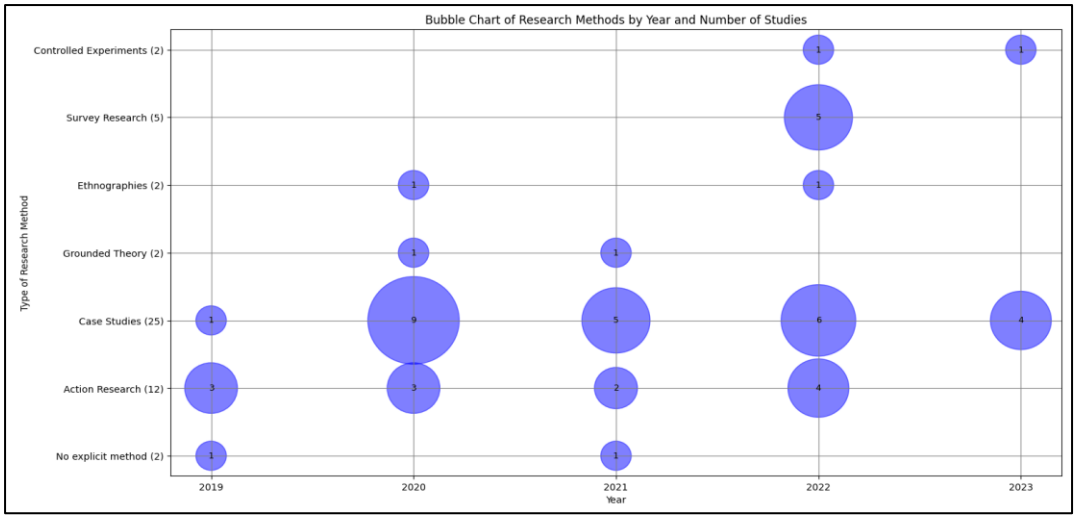


Fig.6. Breakdown of research methods type by year

method, particularly peaking in 2020, indicating a preference for in-depth contextual analysis. Action Research was the second most common, with consistent use across the years, highlighting a focus on collaborative problem-solving in Agile environments. Survey Research was employed in several studies to collect quantitative and qualitative data. Less frequently used methods included Ethnographies and Grounded Theory, each appearing in two studies, reflecting a limited yet significant use of qualitative approaches. Controlled Experiments were documented in two studies, suggesting challenges in planning comprehensive experiments within Agile UCD due to its complexity.

For RQ2, the focus is on the types of papers published on Agile UCD. Using Wieringa et al.'s classification system, four paper types were primarily analyzed: Solution, Evaluation, Experience, and Solution and Evaluation. The analysis from 2019 to 2023 revealed that Experience papers were the most published, particularly in 2022, highlighting practical insights and real-world applications of Agile UCD [4]. Solution papers, which propose new methods or improvements, were also consistently present, reflecting ongoing innovation. Evaluation papers, focusing on the effectiveness of Agile UCD practices, were moderately frequent. Validation papers were the least common, indicating less emphasis on strictly validating existing practices compared to other types of research. The distribution shows a strong preference for practical experience and innovation within the Agile UCD research community[3].

For RQ3, the question explores the benefits proposed by research on integrating Agile and UCD methodologies. An analysis of 50 research paper abstracts identified several key benefits, including enhanced user experience and usability, which were the most frequently mentioned [29]. These studies emphasize how iterative design processes and user-centered evaluations in Agile frameworks improve usability and overall user experience. Increased collaboration and stakeholder involvement were also highlighted, fostering better communication and engagement among team members. Enhanced adaptability and flexibility were noted, showing the integration's ability to handle changing requirements effectively. Other benefits included improved development practices and increased user engagement and satisfaction, reflecting the positive impact of Agile and UCD integration on software development processes.

Overall, these questions and their analyses provide a comprehensive overview of the research methods, types of studies, and benefits associated with Agile UCD, underscoring the dynamic and evolving nature of this research field.

6. CONCLUSION

In conclusion, this systematic mapping study on Agile UCD in software engineering research offers a comprehensive overview of the field's current trends and methodologies. The variety of HCI techniques, notably "Hybrid" and "Collaborative Design," showcase a committed and ongoing effort to merge Agile and UCD principles effectively. This commitment is also evident in the predominance of Case Studies, highlighting a preference for applying these methods in real-world scenarios.

The types of publications we have analysed—ranging from experiential and solution-focused papers to evaluations and validations—illustrate a community eager to share insights, forge new paths, and rigorously test the results of Agile UCD initiatives. This mix not only enriches the knowledge pool but also paves the way for future research by spotlighting successful approaches and pinpointing areas ripe for further inquiry.

The benefits highlighted throughout our research point to substantial improvements in user engagement and satisfaction, directly attributable to the integration of Agile and UCD methodologies. These advancements go hand in hand with enhanced development practices and increased collaboration, both of which are instrumental in elevating the efficiency and output quality of software projects.

Looking to the future, it is clear that combining Agile and UCD is crucial for developing effective, user-focused software. We must keep up with these trends and continually refine our methods to meet the changing needs of users and the dynamics of the software development

industry.

7. REFERENCES

[1] Pakhtusova, Y., Megha, S., & Askarbekuly, N. (2021). A case study on combining agile and User-Centred design. In Frontiers in Software Engineering: First International Conference, ICFSE 2021, Innopolis, Russia, June 17–18, 2021, Revised Selected Papers 1 (pp. 47-62). Springer International Publishing.

[2] Manifesto for Agile Software Development. Agilemanifesto.org, 2019. https://agilemanifesto.org/.

[3] Mao, J.Y., Vredenburg, K., Smith, P.W., Carey, T.: The state of User-Centred design practice. Communications of the ACM 48(3), 105–109 (March 2005)

[4] Silva da Silva, T., Silveira, F. F., Silveira, M. S., Hellmann, T., & Maurer, F. (2015). A systematic mapping on agile UCD across the major agile and HCI conferences. In Computational Science and Its Applications--ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part V 15 (pp. 86-100). Springer International Publishing.

[5] Jönsson, E.: Agile Development and User-Centred Design - A Case Study at Sony Mobile Communications AB, Lund University (2013)

[6] Zorzetti, M., Moralles, C., Salerno, L., Pereira, E., Marczak, S., & Bastos, R. (2021). Adopting Agile software development combined with User-Centred design and lean startup: a systematic literature review on maturity models. In International Conference on Enterprise Information Systems (pp. 517-541). Springer, Cham.

[7] M. Bastarrica, G. Espinoza, J. Sánchez, Implementing agile practices: The experience of Tsol, in: Proceedings of the International Symposium on Empirical Software Engineering and Measurement, ACM, Oulu, Finland, 2018, pp. 1–10, http://dx.doi.org/10.1145/3239235.3268918.

[8] K. Vilkki, When agile is not enough, in: Proceedings of the International Conference on Lean Enterprise Software and Systems, Springer, Helsinki, Finland, 2010, pp. 44–47, http://dx.doi.org/10.1007/978-3-642-16416-3_6.

[9] Da Silva, T.S., Martin, A., Maurer, F., Silveira, M.: User-Centred Design and Agile Methods: A Systematic Review, Proc. - 2011 Agil. Conf. Agil. 2011, no. September, pp. 77--86 (2011)

[10] Interaction Design Foundation, https://www.interaction-design.org/literature/topics/userCentred-design

[11] Dahl, A. Agile/UX Integration: how user experience related practices and processes are integrated with Agile development processes in real-world projects. 2012.

[12] Ferreira, J. Agile Development and UX Design: Towards Understanding Work Cultures to Support Integration. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, (2012), 608-615.

[13] Varatharajah, T. (2019). Integrating UCD with Agile Methods: From the perspective of UX-Designers.

[14] [14] Silva da Silva, T., Martin, A., Maurer, F. and Silveira, M. User-Centred Design and Agile Methods: A Systematic Review. 2011 AGILE Conference, (2011).

[15] da Silva, T., Silveira, M., de O. Melo, C. and Parzianello, L. Understanding the UX Designer's Role within Agile Teams. Design, User Experience, and Usability. Design Philosophy, Methods, and Tools, (2013), 599-609.

[16] A. Ogunyemi, D. Lamas, M. Lárusdóttir, and F. Loizides, "A systematic mapping study of HCI practice research," International Journal of Human-computer Interaction, vol. 35, no. 16, pp. 1461–1486, Nov. 2018, doi: 10.1080/10447318.2018.1541544.

[17] V. G. Ferreira and E. D. Canedo, "Design sprint in classroom: exploring new active learning tools for project-based learning approach," Journal of Ambient Intelligence & Humanized Computing/Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 3, pp. 1191–1212, Mar. 2019, doi: 10.1007/s12652-019-01285-3.

[18] A. De Oliveira Sousa and N. M. C. Valentim, "Prototyping Usability and User Experience," SBQS '19: Proceedings of the XVIII Brazilian Symposium on Software Quality, Oct. 2019, doi: 10.1145/3364641.3364667.

[19] P. Quezada, R. Cueva, and F. Paz, "A Systematic Review of User-Centred Design Techniques applied to the design of mobile application user interfaces," in Lecture notes in computer science, 2021, pp. 100–114. doi: 10.1007/978-3-030-78221-4_7.

[20] A. Hinderks, F. J. D. Mayo, J. Thomaschewski, and M. J. Escalona, "Approaches to manage the user experience process in Agile software development: A systematic literature review," Information and Software Technology, vol. 150, p. 106957, Oct. 2022, doi: 10.1016/j.infsof.2022.106957.

[21] A. Leinonen and V. Roto, "Service Design Handover to user experience design – a systematic literature review," Information and Software Technology, vol. 154, p. 107087, Feb. 2023, doi: 10.1016/j.infsof.2022.107087.

[22] L. Alabood, Z. Aminolroaya, D. Yim, O. Addam, and F. Maurer, "A systematic literature review of the Design Critique method," Information and Software Technology, vol. 153, p. 107081, Jan. 2023, doi: 10.1016/j.infsof.2022.107081.

[23] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE 2008, Swinton, UK, pp. 68–77. British Computer Society, UK (2008).

[24] Ouzzani, M., Hammady, H., Fedorowicz, Z., et al. (2016). Rayyan—a web and mobile app for systematic reviews. Systematic Reviews, 5, 210. https://doi.org/10.1186/s13643-016-0384-4.

[25] K.-J. Stol and B. Fitzgerald, "The ABC of Software Engineering Research," ACM Transactions on Software Engineering and Methodology, vol. 27, no. 3, pp. 1–51, Jul. 2018, doi: 10.1145/3241743.

[26] [Ferre, X., & Medinilla, N. (2007). How a human-Centred approach impacts software development. In Proceedings of the 2007 HCII conference (pp. 68–77). Berlin: Springer-Verlag Berlin Heidelberg

[27] Downey, L. L. (2007). Group usability testing: Evolution in usability techniques. Journal of Usability Studies, 2(3), 133–144.

[28] Abiodun Afolayan Ogunyemi, David Lamas, Marta Kristin Lárusdóttir & Fernando Loizides (2018): A Systematic Mapping Study of HCI Practice Research, International Journal of Human–Computer Interaction, DOI: 10.1080/10447318.2018.1541544

[29] Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. Requir. Eng. 11(1), 102–107 (2005)