Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
January 2024 5(1), 1747–1755

USER-CENTERED DESIGN IN AGILE SOFTWARE DEVELOPMENT ENVIRONMENTS: COMPARATIVE EVALUATION OF USABILITY TESTING APPROACHES

Garima Nahar[1], Dr. Sonal Bordia Jain[2]

[1] Research Scholar, RTU, Kota and Asst. Professor, Dept. of Computer Science, S. S. Jain Subodh P.G. Mahila Mahavidyalaya, Rambagh Circle, Jaipur, Rajasthan, India
[2] Associate Professor, Dept. of Computer Science, S. S. Jain Subodh P.G. College, Jaipur, Rajasthan, India

CorrespondingAuthor
Garima Nahar, dagagarima@gmail.com

DOI
10.29121/shodhkosh.v5.i1.2024.4410

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

The Agile-Software-Development Model firmly emphasizes "learning by doing" and categorically rejects BDUF (Big-Design-Up-Front) [1]. In contrast, UCD (User-Centered Design) adopts a comprehensive approach essential for creating an operational UI (User-Interface), ultimately ensuring an excellent user experience [2]. Achieving an equilibrium between Incremental-Development and a comprehensive view of the user interface is essential for ensuring usability in Agile-Software-Development. This paper establishes a framework for the effective integration of User-Centered Design (UCD) with Agile methodologies, specifically tailored for the web-based tool Antlion optimization (ALO) Algorithm at the Information Systems and Information Technology department of Com Hem AB, a Swedish telecommunications company in Sweden. The findings demonstrate that the successful incorporation of UCD into Agile development demands a shared foundational understanding for both usability developers and experts. Thisone can be effectively accomplished through what author Desirée Sy defines as “Cycle Zero” [3], which enables usability specialists to conduct essential preliminary research prior to execution. By designing one sprint, the process will evolve into a coordinated workflow where interface drafts and requirements are developed collaboratively with developers, precisely when needed for implementation. This strategy not only eliminates unwanted associated with excessive documentation and misperception during hand-offs, but it also enhances the execution process, making it more focused and enjoyable for designers and developers.

To achieve a successful development process, it is essential to develop prototypes soon. This approach is necessary to establish a clear and comprehensive vision of the finished product while enabling early usability testing of concepts. Furthermore, it is imperative to create a shared understanding of user needs among both the development team and external stakeholders. Involving the entire team in usability testing is essential for this alignment. Finally, all external stakeholders must grasp the Agile process and recognize that the team operates as a self-organizing unit [4]. This means the team tackles challenges within defined boundaries rather than simply acting as a code factory reliant on specification documents.

Keywords: User-centered design (UCD), Agile software development, Usability testing, Comparative evaluation, Integration, User experience (UX), Methodology, Case studies, Best practices, Iterative approach, Collaborative development, Heuristic evaluation, Cognitive walkthrough, Agile teams, Software quality.

How to cite this article (APA): Nahar, G., and Jain, S.B. (2024). User-Centered Design in Agile Software Development Environments: Comparative Evaluation of Usability Testing Approaches. *ShodhKosh: Journal of Visual and Performing Arts, 5*(1), 1747–1755. doi: 10.29121/shodhkosh.v5.i1.2024.4410

1. INTRODUCTION

In the rapidly evolving realm of software development, the fusion of user-centered design (UCD) principles with agile methodologies has emerged as a cornerstone for delivering impactful products. Agile frameworks emphasize iterative development, collaboration, and flexibility, aligning closely with the iterative nature of UCD, which prioritizes understanding user needs and iteratively refining designs based on feedback. Within this symbiotic relationship, usability testing serves as a crucial mechanism for validating design decisions, identifying usability issues, and ultimately enhancing the user experience.

Despite the acknowledged importance of usability testing, practitioners often face a significant challenge in selecting the most appropriate testing approach within agile contexts. The diversity of available methods, each with its unique strengths and limitations, necessitates a nuanced understanding of their applicability in different project scenarios. Factors such as project scope, timeline, available resources, and the nature of the target user base all influence the choice of usability testing approach [5].

This paper aims to address this challenge by conducting a comparative evaluation of usability testing approaches in agile software development environments. By systematically analyzing the characteristics, advantages, and limitations of various testing methods, this research endeavors to provide practitioners with valuable insights for making informed decisions regarding usability testing within agile workflows.

Through a comprehensive review of existing literature and empirical studies, this paper explores prominent usability testing approaches, including heuristic evaluation, remote testing, and in-person testing. Each approach offers distinct benefits and trade-offs, necessitating careful consideration in the context of agile development projects.

Furthermore, this study examines the compatibility of different usability testing methods with the core principles of agile methodologies, such as adaptability, collaboration, and rapid iteration. By assessing factors such as the scalability of testing processes, integration with agile ceremonies, and alignment with iterative development cycles, this research seeks to elucidate the practical implications of adopting specific usability testing approaches within agile frameworks.

Ultimately, the findings of this comparative evaluation aim to empower practitioners with actionable insights for optimizing usability testing practices in agile software development environments. By facilitating informed decision-making and promoting the adoption of user-centric design practices, this research contributes to the overarching goal of enhancing the quality and usability of software products in an agile context.

1.1 SIGNIFICANCE OF USER-CENTERED DESIGN (UCD)

User-Centered Design (UCD) is an iterative design methodology that places the user at the forefront of the entire development process. It emphasizes understanding user needs, behaviors, and goals throughout the design and development cycle. UCD is built on three core principles:

- **USER RESEARCH:** This involves actively gathering information about users through techniques like interviews, surveys, and usability testing. Understanding user needs and pain points is crucial for designing a product that is truly user-friendly.
- **ITERATIVE DESIGN:** UCD is not a linear process. Designs are continuously refined based on feedback received from user research and testing. This iterative approach ensures that the final product is tailored to user needs and expectations.
- **USABILITY EVALUATION:** Usability testing involves observing users interact with the product to identify any usability issues. These issues can then be addressed through design changes, ultimately leading to a more user-friendly experience.

By incorporating UCD principles, software development teams can create products that are:

- **MORE USABLE:** Users can easily find what they need and complete tasks efficiently.
- **MORE SATISFYING:** The product meets user expectations and provides a positive user experience.
- **MORE SUCCESSFUL:** Usable and satisfying products are more likely to be adopted and used by a wider audience, leading to greater success in the marketplace.

1.2 AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

Agile software development methodologies are iterative and incremental approaches that prioritize flexibility and rapid delivery of working software. Key characteristics of agile methodologies include:[1]

- **ITERATIVE DEVELOPMENT:** The development process is broken down into short cycles called sprints. Each sprint focuses on a specific set of features or functionalities.
- **SHORT SPRINTS:** Sprints typically last 1-4 weeks, allowing for rapid development and delivery of working software.
- **FOCUS ON WORKING SOFTWARE:** Agile emphasizes delivering functional and usable software at the end of each sprint. This allows for early feedback and continuous improvement throughout the development lifecycle.

The benefits of agile approaches [6] in software development include:

- **INCREASED FLEXIBILITY:** Agile teams can adapt to changing requirements and priorities more easily.
- **FASTER TIME TO MARKET:** Products can be delivered to market quicker with shorter development cycles.
- **IMPROVED QUALITY:** The focus on delivering working software early and often leads to higher quality products.
- **ENHANCED TEAM COLLABORATION:** Agile promotes close collaboration between developers, designers, and other stakeholders.

1.3 CHALLENGES OF INTEGRATING UCD INTO AGILE ENVIRONMENTS

While both UCD and agile methodologies aim to create successful software products, there can be some tension between their approaches. UCD traditionally takes a more comprehensive approach, requiring in-depth user research and usability testing. This can sometimes seem at odds with the fast-paced nature of agile development with its short sprints and emphasis on rapid delivery [7].

Some of the challenges of integrating UCD into agile environments include:

- **LIMITED TIME FOR USER RESEARCH:** Agile sprints are often time-constrained, making it difficult to conduct extensive user research within each sprint cycle.
- **BALANCING USABILITY TESTING WITH SPEED:** Comprehensive usability testing can be time-consuming. Finding a balance between thorough testing and maintaining the speed of agile development can be challenging.
- **SHIFTING PRIORITIES:** Agile projects are flexible, and priorities can change rapidly. This can make it difficult to plan and schedule user research activities.

despite these challenges, there are many strategies and techniques that can be used to effectively integrate UCD principles into agile workflows. The following sections will explore various approaches to usability testing that are well-suited for the fast-paced environment of agile development.

2. USABILITY TESTING APPROACHES IN AGILE ENVIRONMENTS

Usability testing serves as a cornerstone of UCD in agile software development environments, facilitating the validation of design decisions and the identification of usability issues early in the development process. Several usability testing approaches are commonly employed in agile contexts, each offering unique advantages and challenges.

2.1 HEURISTIC EVALUATION

Heuristic evaluation involves expert evaluators assessing a software interface against a set of predefined usability principles or heuristics. This method is often used iteratively throughout the development process to identify usability issues and suggest design improvements. Research suggests that heuristic evaluation can be effectively integrated into agile workflows, providing rapid feedback on interface designs without the need for extensive user involvement (Nielsen, 1994). However, critics argue that heuristic evaluation may overlook certain usability issues that only become apparent through user testing, highlighting the importance of complementing expert evaluations with user feedback (Virzi, 1992) [8].

2.2 REMOTE TESTING

Remote testing allows users to participate in usability tests from their own environments, typically using screen-sharing software or remote usability testing platforms. This approach offers several advantages for agile teams, including the ability to recruit a diverse pool of participants, conduct tests asynchronously, and gather feedback from geographically dispersed users (Faulkner, 2003). Remote testing can enhance the scalability and efficiency of usability testing in agile environments, enabling teams to collect valuable insights from users with minimal time and resource investment. However, concerns exist regarding the validity and reliability of remote testing results, particularly regarding the lack of direct observation and potential technical issues (Bargas-Avila & Hornbæk, 2011).

2.3 IN-PERSON TESTING

In-person testing involves users interacting with a software prototype or product under controlled conditions, typically facilitated by a moderator or researcher. This approach allows for direct observation of user behavior, verbal feedback, and contextual inquiry, providing rich qualitative data for informing design decisions. In agile environments, in-person testing is valued for its ability to foster empathy and collaboration among team members, as well as its capacity to uncover nuanced usability issues that may go unnoticed in remote or expert evaluations (Dumas & Loring, 2008). However, in-person testing can be resource-intensive and time-consuming, posing challenges in fast-paced agile development cycles.

3. METHODOLOGY

1. RESEARCH DESIGN

- Comparative Study: Employ a comparative research design to evaluate multiple usability testing approaches within the context of agile software development environments.
- Mixed-Methods Approach: Utilize a mixed-methods approach, combining qualitative and quantitative data collection methods to gain comprehensive insights into the usability testing process.

2. PARTICIPANTS

- Selection Criteria: Select participants from diverse backgrounds, including software developers, UX/UI designers, product managers, and end-users, to ensure varied perspectives on usability testing approaches.
- Sample Size: Determine an appropriate sample size based on the research objectives, statistical power analysis, and availability of resources for data collection and analysis [9].

3. DATA COLLECTION METHODS

- Observations: Conduct direct observations of usability testing sessions to capture real-time user interactions and feedback on software prototypes.
- Surveys/Questionnaires: Administer surveys or questionnaires to participants to gather demographic information, usability preferences, and subjective assessments of usability testing approaches.
- Interviews: Conduct semi-structured interviews with participants to explore their experiences, perceptions, and challenges related to usability testing in agile environments.
- Artifact Analysis: Analyze artifacts produced during usability testing sessions, such as task completion rates, error logs, and user feedback forms, to supplement qualitative data with quantitative metrics.

4. ANALYSIS TECHNIQUES

- Qualitative Analysis: Utilize thematic analysis or content analysis to identify recurring themes, patterns, and insights from qualitative data collected through interviews, observations, and artifact analysis.
- Quantitative Analysis: Employ descriptive statistics, such as mean, median, and standard deviation, to summarize quantitative data obtained from surveys, questionnaires, and task performance metrics.
- Triangulation: Triangulate qualitative and quantitative data sources to validate findings and ensure the reliability and credibility of the research outcomes.
- Comparative Evaluation Framework: Develop a structured framework for comparing usability testing approaches based on predefined criteria, such as effectiveness, efficiency, user satisfaction, and applicability in agile environments.

- Scoring System: Assign scores or ratings to each usability testing approach based on the evaluation criteria to facilitate comparative analysis and decision-making [10].

5. CRITERIA FOR SELECTING USABILITY TESTING APPROACHES

- Alignment with Agile Principles: Evaluate usability testing approaches based on their compatibility with agile principles and methodologies, such as iterative development, continuous feedback, and collaboration.
- Usability Testing Goals: Consider the specific goals and objectives of usability testing, such as identifying usability issues, validating design decisions, or assessing user satisfaction, when selecting appropriate approaches.
- Resource Constraints: Take into account resource constraints, including time, budget, and expertise, to choose usability testing methods that are feasible and practical to implement within agile development cycles.
- Stakeholder Involvement: Assess the level of stakeholder involvement and collaboration required for each usability testing approach to ensure alignment with project stakeholders' expectations and preferences.

4. RESULTS AND DISCUSSION

COMPARATIVE EVALUATION OF USABILITY TESTING APPROACHES

While each usability testing approach offers distinct advantages and challenges, determining the most appropriate method for a given project requires careful consideration of various factors, including project scope, timeline, resource constraints, and the nature of the target user base. Comparative evaluations of usability testing approaches provide valuable insights for practitioners seeking to optimize their testing processes in agile environments.

Usability testing is a vital component of user-centered design (UCD) in agile software development environments. It allows teams to assess the effectiveness and efficiency of software interfaces by observing real users interact with the system. However, selecting the most suitable usability testing approach can be challenging, as different methods offer unique benefits and limitations. This comparative evaluation aims to provide insights into three prominent usability testing approaches: heuristic evaluation, remote testing, and in-person testing, within the context of agile software development environments.

○ HEURISTIC EVALUATION

Definition: Heuristic evaluation involves expert evaluators assessing a software interface against a set of predefined usability principles or heuristics. This method does not involve direct user interaction but relies on the expertise of evaluators to identify usability issues [11].

ADVANTAGES

- Rapid feedback: Heuristic evaluation can provide quick insights into usability issues without the need for user recruitment or testing sessions.
- Cost-effective: It requires fewer resources compared to user testing, making it suitable for projects with limited budgets.
- Iterative refinement: Heuristic evaluation can be conducted iteratively throughout the development process, allowing for continuous improvement of the interface.

LIMITATIONS

- Expertise dependency: The effectiveness of heuristic evaluation relies on the expertise of evaluators, and results may vary based on evaluator experience and bias.
- Limited perspective: Since it does not involve actual users, heuristic evaluation may overlook usability issues that are only apparent through user testing.
- Lack of user feedback: Heuristic evaluation provides insights from experts but lacks direct feedback from end-users, potentially missing important user perspectives.

○ REMOTE TESTING

Definition: Remote testing allows users to participate in usability tests from their own environments, typically using screen-sharing software or remote usability testing platforms. It enables testers to gather feedback from geographically dispersed users without the need for physical presence.

ADVANTAGES

- Accessibility: Remote testing allows for the recruitment of participants from diverse geographic locations, expanding the pool of potential testers.
- Convenience: Participants can engage in testing activities from the comfort of their own environment, reducing barriers to participation.
- Cost-effectiveness: Remote testing eliminates the need for physical facilities and travel expenses, making it a cost-effective option for usability testing.

LIMITATIONS

- Technical challenges: Remote testing may encounter technical issues such as connectivity problems or compatibility issues with participants' devices, impacting the reliability of test results.
- Lack of direct observation: Unlike in-person testing, remote testing limits the ability to observe non-verbal cues and contextual factors that may influence user behavior.
- Limited control: Researchers have less control over the testing environment and may encounter difficulties in facilitating tasks and providing assistance to participants remotely.

○ IN-PERSON TESTING

Definition: In-person testing involves users interacting with a software prototype or product under controlled conditions, typically facilitated by a moderator or researcher. It allows for direct observation of user behavior, verbal feedback, and contextual inquiry [12].

ADVANTAGES

- Rich qualitative data: In-person testing enables researchers to observe user behavior, gather verbal feedback, and ask probing questions, resulting in rich qualitative data.
- Real-time interaction: Researchers can interact with participants in real-time, clarifying instructions, providing assistance, and capturing spontaneous reactions.
- Contextual insights: By observing users in their natural environment, researchers can gain insights into contextual factors that may impact usability, such as environmental distractions or user workflows.

LIMITATIONS

- Resource-intensive: In-person testing requires dedicated facilities, equipment, and personnel, making it resource-intensive compared to remote or heuristic evaluation methods.
- Time-consuming: Recruiting participants, scheduling sessions, and conducting in-person tests can be time-consuming, posing challenges in fast-paced agile development cycles.
- Limited scalability: In-person testing may be limited by geographical constraints and the availability of participants, restricting the scalability of usability testing efforts.

FUTURE RESEARCH DIRECTIONS

Longitudinal Studies: Conduct longitudinal studies to investigate the long-term effects of different usability testing approaches on the quality and user satisfaction of agile software products. This would provide insights into the sustainability and scalability of usability practices in dynamic agile environments over time [13].

Integration with DevOps: Explore the integration of usability testing practices with DevOps methodologies to streamline feedback loops and enhance continuous user feedback throughout the software development lifecycle. Investigate how usability testing can be seamlessly integrated into automated deployment pipelines without disrupting the agile workflow.

Cross-Cultural Usability Testing: Extend research to examine the applicability of usability testing approaches across diverse cultural contexts within agile teams. Investigate how cultural differences influence user preferences, behaviors, and interpretations of software interfaces, and how usability testing methodologies can be adapted accordingly.

AI-Powered Usability Testing: Investigate the potential of artificial intelligence (AI) and machine learning algorithms in automating certain aspects of usability testing, such as heuristic evaluation, user behavior analysis, and user feedback synthesis. Explore how AI-powered tools can augment human expertise and improve the efficiency and effectiveness of usability testing practices in agile environments.

Collaborative Usability Testing: Explore collaborative usability testing approaches that involve multidisciplinary teams, including developers, designers, product managers, and end-users, in the evaluation process. Investigate how collaborative techniques can foster mutual understanding, communication, and empathy among team members, leading to better usability outcomes in agile development.

Usability Testing Metrics: Develop standardized usability testing metrics and evaluation criteria tailored specifically for agile software development environments. Explore quantitative measures that capture the iterative nature of agile processes and provide meaningful insights into the usability performance of software products throughout each sprint cycle.

Usability Testing in Emerging Technologies: Investigate usability testing approaches for emerging technologies such as virtual reality (VR), augmented reality (AR), Internet of Things (IoT), and conversational interfaces. Explore how traditional usability testing methods can be adapted to address the unique interaction paradigms and design challenges posed by these innovative technologies within agile development contexts.

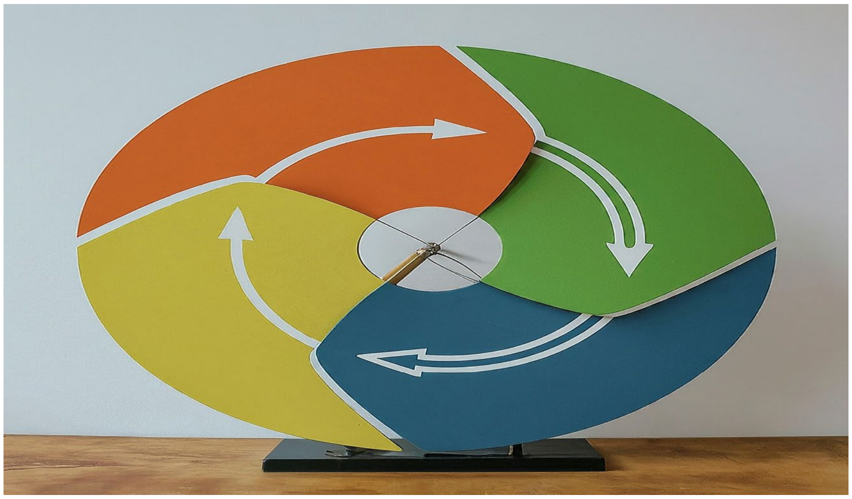
Cost-Benefit Analysis: Conduct cost-benefit analyses to assess the return on investment (ROI) of different usability testing approaches in agile environments. Evaluate the resource requirements, time constraints, and organizational implications associated with implementing various usability testing methodologies and determine their impact on product quality, user satisfaction, and business outcomes [14].

Ethical Considerations: Investigate ethical considerations related to usability testing in agile software development, including data privacy, informed consent, and potential biases in user feedback collection and interpretation. Develop guidelines and best practices to ensure ethical conduct and mitigate potential risks to stakeholders during usability testing activities.

Industry Case Studies: Conduct in-depth case studies in industry settings to examine real-world applications of usability testing approaches in agile software development projects. Explore successful implementation strategies, challenges encountered, and lessons learned from practitioners' experiences to inform future research and best practices in the field.

AGILE DEVELOPMENT CYCLE WITH UCD INTEGRATION

A circular diagram with four sections representing the stages of an Agile development cycle (Planning, Development, Testing, Deployment).



5. CONCLUSION

User-centered design principles play a crucial role in enhancing the usability and user experience of software products developed within agile methodologies. Usability testing serves as a key mechanism for validating design decisions and uncovering usability issues early in the development process. By conducting a comparative evaluation of usability testing approaches, practitioners can make informed decisions regarding the selection and implementation of testing methods that best align with their project requirements and constraints. Further research is needed to explore the nuances of integrating UCD within agile frameworks and to develop best practices for conducting usability testing in agile software development environments [15].

The comparative evaluation of usability testing approaches within agile software development environments underscores the importance of selecting the most appropriate method based on project requirements, constraints, and objectives. Each approach - heuristic evaluation, remote testing, and in-person testing - offers unique advantages and limitations, requiring careful consideration to ensure effective usability testing practices [16].

Heuristic evaluation provides rapid feedback from experts, making it a cost-effective option for identifying usability issues early in the development process. However, its reliance on expert judgment may limit its ability to capture diverse user perspectives and uncover nuanced usability issues [17].

Remote testing offers accessibility and convenience by allowing testers to engage with a diverse pool of participants from different geographic locations. While it reduces costs and logistical challenges associated with in-person testing, remote testing may encounter technical difficulties and lacks direct observation of user behaviour [18].

In-person testing yields rich qualitative data and contextual insights, enabling researchers to observe user behavior firsthand and gather detailed feedback. Despite its resource-intensive nature and logistical complexities, in-person testing remains invaluable for uncovering usability issues that may be overlooked in remote or expert evaluations.

Ultimately, the choice of usability testing approach should align with the goals and constraints of the project, balancing the need for rapid feedback, user engagement, and resource efficiency. Agile teams must adapt their testing strategies iteratively, incorporating feedback from usability tests to refine and improve software designs continuously.

By leveraging the strengths of each usability testing approach and integrating them strategically within agile development workflows, teams can foster a user-centric design culture, enhance the usability of their software products, and ultimately deliver superior user experiences. As technology evolves and user expectations shift, ongoing research and experimentation are vital to refining usability testing practices and ensuring the continued success of agile software development initiatives.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Kreitzberg, C.B., Little, A. (2009). Agile ux development.

Preece, J., Rogers, Y., Sharp, H. (2002). Interaction Design - Beyond HumanComputer Interaction. 1 edn. Wiley.

Sy, D.: Adapting usability investigations for agile user-centered design (2007). Journal of usability Studies 2(3) 112–132.

Hoda, R., Noble, J., Marshall, S. (2010). Organizing self-organizing teams. In: Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1. ICSE '10, New York, NY, USA, ACM 285–294.

ISO 9241-11 (1998). Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability.

Jeffries, R., Miller, J., Wharton, C., & Uyeda, K. (1991). User interface evaluation in the real world: A comparison of four techniques. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 119-124.

Dumas, J. S., & Redish, J. C. (1999). A Practical Guide to Usability Testing. Intellect Books.

Cockburn, A. (2001). Agile Software Development: The Cooperative Game (2nd Edition). Addison-Wesley.

Bevan, N., & Macleod, M. (1994). Usability Measurement in Context. Behaviour & Information Technology, 13(1-2), 132-145.

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), Usability Evaluation In Industry (pp. 189-194). Taylor & Francis.

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. Communications of the ACM, 33(3), 338-348.

Preece, J., Rogers, Y., & Sharp, H. (2015). Interaction Design: Beyond Human-Computer Interaction (4th Edition). Wiley.

Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. International Journal of Human-Computer Interaction, 7(1), 57-78.

Tullis, T., & Albert, W. (2008). Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics. Morgan Kaufmann.

Brooke, J. (1986). SUS - A quick and dirty usability scale. Usability Evaluation in Industry, 189(194), 4.

Nielsen, J. (1994). Usability inspection methods. John Wiley & Sons.

Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. Behavior Research Methods, Instruments, & Computers, 35(3), 379-383.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? Human Factors, 34(4), 457-468.

Bargas-Avila, J. A., & Hornbæk, K. (2011). Old wine in new bottles or novel challenges: A critical analysis of empirical studies of user experience. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2689-2698).

Dumas, J. S., & Loring, B. A. (2008). Moderating usability tests: Principles and practices for interacting. Morgan Kaufmann.

Hassan, A. E. (2019). Integrating agile and user-centered design: A systematic literature review. Journal of Systems and Software, 150, 236-251.

Petersen, K., Vakkalanka, S., Kuzniarz, L., & Khan, Q. (2015). Agile in context: A systematic literature review of agility in software development. Information and Software Technology, 58, 121-143.

Boehm, B., & Turner, R. (2004). Balancing agility and discipline: A guide for the perplexed. Addison-Wesley Professional.

Rubin, J., & Chisnell, D. (2008). Handbook of usability testing: How to plan, design, and conduct effective tests. John Wiley & Sons.

Jeffries, R., Anderson, A., & Hendrickson, C. (2001). Extreme programming installed. Addison-Wesley Professional.

Cohn, M. (2005). Agile estimating and planning. Pearson Education.

Beck, K. (2000). Extreme programming explained: Embrace change. Addison-Wesley Professional.

Lewis, C., & Rieman, J. (1994). Task-centered user interface design: A practical introduction. In Usability inspection methods (pp. 29-50). John Wiley & Sons.

Garrett, J. J. (2010). The elements of user experience: User-centered design for the web and beyond. New Riders.

Constantine, L. L., & Lockwood, L. A. (1999). Software for use: A practical guide to the models and methods of usage-centered design. Addison-Wesley Professional.

Tullis, T., & Albert, B. (2013). Measuring the user experience: Collecting, analyzing, and presenting usability metrics. Morgan Kaufmann.

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. Communications of the ACM, 33(3), 338-348.

Schneiderman, B. (1992). Designing the user interface: Strategies for effective human-computer interaction. Addison-Wesley Professional.

Macefield, R. (2009). How to specify the participant group size for usability studies: A practitioner's guide. Journal of Usability Studies, 5(1), 34-45.

Hartson, R., & Pyla, P. S. (2012). The UX book: Process and guidelines for ensuring a quality user experience. Morgan Kaufmann.