

Critical Success Factors for Implementing Empathic Design in Agile Teams

Annie Maré^{1[0000-0002-0890-6945]} and Marié Hattingh^{2[0000-0003-1121-8892]}

^{1,2} University of Pretoria, Private Bag X20, Hatfield, 0028
¹mareannie@gmail.com, ²marie.hattingh@up.ac.za

Abstract. Agile software teams focus on achieving a working product as fast as possible. This, however, is often accomplished to the detriment of user-centered design principles. Although several tools have been developed that enable agile teams to implement empathic designs in their projects, literature indicates that user experience designers struggle to influence software development teams, while developers find it challenging to collaborate with designers and remaining agile. This study serves to identify critical success factors (CSF) for the implementation of empathic design in agile software development teams. An interpretive case study was performed on an agile software development team to identify these CSF for implementing empathic design. This study found that client buy in, and the implementation of a hybrid agile method are important to the success of implementing empathic design.

Keywords: HCI, Usability, User-centered design, Agile, Empathic design, Critical success factors

1 Introduction

The core of agile software development is its ability to respond fast to any changes in the software development environment. Developers that use agile development methodologies seek to gain knowledge from the existence of the product, thus the need to fail fast so that faults in the product can be identified and improved upon [1]. Agile software development focuses more on producing functional software rather than on documentation and processes [1-4].

Praised for its lightweight, fast-moving pace, agile teams produced software in iterative releases as opposed to publishing it once the entire system was fully functional [1-3, 5]. Each mini release needs to be a stable version of the software where more features can be built. These mini releases are usually done in short sprint cycles during which the development team focuses on releasing new features in each sprint [1, 3].

Unfortunately, end users are often not part of the development process or are involved in unrealistic and unsustainable ways such as acting as “site end users” or end-user representatives [4, 6].

Expected end users, or users that fall within the same demographic as the expected end users can be interviewed in focus groups or individually. During these sessions

Smuts, Hanlie & Hinkelmann, Knut (2024). Proceedings of the 4th Conference Society 5.0 - Innovation for Sustainable and Inclusive Social Good, Volume 2, Mauritius, 26-28 June 2024, DOI: 10.5281/zenodo.11612759

DOI: 10.5281/zenodo.11660995

the UX researcher may ask the participants to perform card sorting exercises, participatory design exercises and usability tests on any existing software. Rigorous user-centred methods of field investigation can be time consuming, a stark contrast to the fast paced, quick release standards of the agile process [4, 7].

User-centred design (UCD) is a design philosophy that emphasises the user's needs throughout the development process [8]. The core principle involves that the intended end user is consulted continuously throughout the product life cycle, using the above-mentioned user-centred methods of field investigation. UX designers use UCD, aiming to identify and document as much of the requirements beforehand as possible, allowing for a holistic view of the environment in which the product functions so that a useful solution can be created [9]. Once a product has been released, field research is conducted by observing how users interact with the product, whereafter changes are made to increase the accessibility of the product. Alterations to the design of the product, based on the feedback received from conducting field research, are introduced in iterative stage [4, 8], and design translations must be facilitated between designers and developers to ensure that developers understand the reasons for the alterations.

Several techniques have been created that help to facilitate empathy during the design process, such as user-centred design, human-centred design, co-design (Co-D) and participatory design (PD) [10]. These design techniques enable empathic design by allowing the designer to take the user's perspective and practice cognitive empathy. Designers are encouraged to envision the user's thoughts, feelings, and needs to identify the requirements of the software solutions [10].

Although these techniques are well established for designers, there seems to be a lack of methods for enhancing empathy towards the end user in the workplace in general [11]. The goal of design is to create a useful product for the intended user; the extent of the design's success can only be proven by the users themselves [11, 12]. The use of design thinking allows for creating a technically viable software project that meets the user's overall expectation while remaining economically feasible to execute [13]. This paper asks the main research question: What are the Critical Success Factors for implementing empathic design in an agile software development environment?

In section 2, a literature review on software methodologies, user experience design, empathic design and aligning empathic design with agile are discussed. Section 3 the research methodology followed in this study is described followed by the research findings and conclusion in Sections 4 and 5 respectively.

2 Literature review

2.1 Software Development Methodologies

There are several Software Development Life Cycle (SDLC) models that are used to manage the phases of development such as planning, analysis and implementation. Two of the SDLC models that are relevant to this research paper are the Waterfall and Agile methodologies.

The Waterfall method consists of the sequential development of software, where the output of a previous phase will serve as input for future phases [14-16]. Each phase of the development process must be completed before the next phase of development can begin. The Waterfall method is the most traditional project management methodology that allows team members to work linearly towards an end-goal. Every member of the team has a set goal to achieve; and the estimations and expectations do not change over time [14]. Testing is only carried out once the software has been fully developed, which causes defects to be detected very late in the software life cycle [15].

Agile software methodologies were developed to respond quickly to the ever-changing environment in which software is created [2, 14, 17-19]. Although agile is highly compatible with user-centred design techniques, the focus on creating usable products usually diminishes throughout the project timeline, and the focus is shifted towards producing working versions of the product as quickly as possible [2, 3, 8, 20, 21]. The Agile methodology's iterative and incremental approach allows for effective system requirements management and aligning the product with customer needs [19, 21, 22].

Faced with the benefits and challenges of both agile and the waterfall methodology, software teams have recently started implementing a hybrid agile methodology [21, 23, 24]. Hybrid agile promotes the combination of plan-based development models such as the waterfall methodology, with the agile software development methodology. This method provides a range of benefits including improved project quality, faster time to market and better resource allocation accuracy due to better workload estimation [21, 23].

2.2 User experience (UX) and User-Centred design (UCD)

User experience is defined in ISO 9241-210 as 'a person's perceptions and responses that result from the use or anticipated use of a product, system or service' [18]. Hinderks, et al. [18] conclude that the user experience is thus a holistic concept that encompasses the physical, emotional, and cognitive reactions a user has when interacting with a product. This experience can happen before, during or after the use of the product.

User-centeredness or user-centred design is a design practice where designers follow a process of focusing on usability, user goals and desires throughout the development stages of the software, as well as later during the system life cycle [2, 25, 26]. User-centred design is defined as an approach for developing interactive systems to make a product usable and useful, with a particular focus on the user to understand their experiences and requirements [27]. It highlights the need to understand the human factors and usability techniques when developing software.

User experience (UX) designers fulfil a range of roles in the product development and design process. Their influence ranges from conducting user research, creating prototypes, designing user interfaces, and even specialising in how users interact with company copy (content created by a company to increase brand awareness that is meant to persuade customers to act). UX designers usually practice UCD as it is the

most effective way to design products that satisfy all the user's needs while also being delightful to use [28]. UX designers intend to empathise with users in various ways, such as through user interviews, observations, and personas. Personas are fictional characters that embody the characteristics and traits of the intended end user that allows designers to design products that are more aligned with the user's goals and needs [25].

2.3 Empathic Design

Empathy is widely researched in many fields, each focusing on different aspects of the broader term. Empathy is defined as “the ability to understand the feeling of others” [29]. Empathy aims to construct a mutual understanding of how a person perceives an experience [11]. Empathy has a critical impact on design thinking [43]. Kouprie and Visser [30] argue that by using empathy as a core part of the design process, the designer is moved away from solving rational issues and focuses more on solving for personal experience and user context. Perspective taking is a skill that designers need to enhance to create delightful, impactful and valuable products [10, 31]. Empathy should become part of an organisation's knowledge construction as empathetic insights can be gained from interactions with people regarding the technology used, knowledge about user needs, user language, cultural symbols etc. Designers and developers should learn to adjust their own perspectives in favour of the perspective of the user [11, 26].

Designers use different tools to increase empathy with the user while designing. These tools include the use of personas, which allow designers to empathise with fictional characters, similar to how a reader would empathise with characters in a storybook [25].

2.4 Aligning User centred design and Software development

Due to the functioning of each role, designers and developers look at problems from different perspectives. UCD aims to produce the whole user experience while developers divide the work into smaller releases [22]. UCD works with a more holistic view, whereas developers break down problems into modular sections. This causes a disjoint between the mentalities of designers and developers that are extremely difficult to manage [9, 26, 28].

Designers often perform the role of producing ad-hoc solutions, reviewing and giving feedback on products etc., the primary measure of success being “working software”. This can cause strain on the relationship between developers and designers [28]. Studies have shown that there is a substantial misalignment of attitudes and work practices between developers and designers, and successful cooperation between these two roles ties in strongly with the organisational culture and standard decision-making process used by the organisation [26, 28]. Elsbach and Stigliani [32] found that organisations that successfully implement design thinking tools usually include collaboration across teams from different departments, including marketing.

Often in agile spaces, UCD is only considered at the later stages of development or during the testing of the software [2, 26]. It was found that in agile software development methods, sprints became too short if user testing and usability testing increased [33].

Several researchers suggest a dual-track agile software development method. Using this method, UX designers work one sprint ahead of developers to ensure the designs are within the scope and that validated designs are delivered to the development team so that a tangible form of the proposed solution can be developed [3, 9, 33, 34]. The problem with this approach is that a single sprint usually does not allow the designer to understand the features they are addressing in the context of the business [20, 28]. Ideally, the UX team should conduct small-scale user acceptance tests throughout the development process to ensure that any change made to the designs during the development process still fits the user's expectations [34].

It is thus essential for development teams to have regular alignment meetings to ensure that the designs align with development capacity while still ensuring that the user's needs will be met [9, 28]. It is also critical in the UX process for a diverse group to evaluate the consequences of the suggested solution from their perspective and skillset to ensure that the team is aligned to achieve the end goal [9, 28].

2.5 Challenges between agile methodologies and user experience

The adoption of agile methodology has repercussions on usability as it prefers a minimalistic, "address it as it happens" approach over an in-depth analysis of the user requirements [4, 22].

The problem with the practices mentioned above is that it does not cater to cases where integration does not occur constantly or occurs unpredictably. It also does not consider that agile teams rarely follow the methodologies accurately in the practice [4, 22]. Usability designers that are fully integrated with the development team may lose track of the UX goals and the mere presence of a UX designer in meetings and planning sessions may not be enough to ensure adequate thought is given to the user experience [4].

Studies found that in Agile organisation UX design does not receive the necessary support and UX designers often not part of the development team, but are responsible for delivering designs for multiple development teams, reducing their effectiveness and efficiency [18, 26]. The most commonly used usability methods in agile software development environments include: heuristic evaluations, fast prototyping, individual inquiry and formal tests [18].

At the pace that agile teams need to release new features to the existing software, there is often not enough time to do adequate user testing [8, 18, 34]. There are often not enough pieces of the puzzle to effectively test a feature during a sprint, and user tests are often moved back to the end of the project. At this point, it is far too late for the UX researcher's findings to have any measurable impact on the feature being released and is often only added to later sprints and, in some cases, never addressed [34].

Agile teams are often averse to documentation since it is such a people-focused development methodology that prefers just-in-time requirement solicitations over in-depth investigations [4, 22, 35]. In the development team's fast-paced, product delivery mindset, documentation is often left on the back burner, then only appreciated when system maintenance needs to be done [14, 21, 35].

3 Methodology

The research philosophy implemented for this study was interpretivism, as it aims to explain the views and experiences of a group of people within a specific social setting. A qualitative research method was used for data collection as it allows for the analysis of social phenomena and the collection of non-numeric data. An interpretive case study was conducted to gain an in-depth understanding of the agile organisation that was analysed. Twenty-one participants were chosen for this case study based on their relevance to the study through the selective sampling method, and data was collected by conducting semi-structured interviews with the selected participants. A thematic analysis is used to identify common themes and patterns within the qualitative data that was collected. Data was collected from Organisation X.

Organisation X is a small to medium software development company that offers analysis and design of a system and the development and maintenance of systems. The organisation employs a variety of roles from software developers, information designers, user experience designers, business analysts and project managers. The organisation works with various clients on various projects, each with a unique user base and system requirements.

4 Research Findings

The data collected was thematically analysed to assist to answer the research questions. Five types of challenges were identified, communication challenges, project constraints challenges, developer challenges, process challenges and design challenges. The agile principles that impede the implementation of empathic design were identified, and a list of critical success factors that were suggested by the interviewees were summarised. The themes and their subthemes are visually represented in Figure 1.

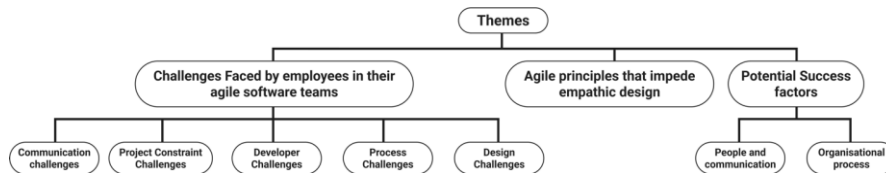


Fig. 1. Codes and related themes from interviews

4.1 Challenges faced by the Interviewees in their Agile Teams

The challenges faced by the interviewees were divided into five subgroups, namely: Communication challenges, Project constraints, Challenges faced by developers, Process challenges, and Design challenges.

Communication Challenges. In the communications challenge category, one of the biggest challenges faced by the interviewees in terms of implementing empathic design was client buy in. Clients tended to value the output of development higher than the process of designing empathic software. One of the developers that were interviewed, when referring to why their client does not buy into the design process on their project, stated that ..."business does not necessarily see the value in a certain feature being developed in a one way or a different way". One of the design team leads stated that they try to persuade their client that ... "It's actually wasting your money to try and build a system where you don't keep the user in mind." This aligns with findings by Bongiovanni and Louis [36] that found that the language used by designers and consultants differ and affect the buy in from clients. This is also in contradiction to agile principle: "Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done" as creates a barrier for designers to perform their work effectively.

Another challenge with implementing empathic design is the cost associated with user testing and the availability of user representative to test the software's usability. Clients are often unwilling to pay for recruitment of users for either virtual or in person user tests. As stated by one of the design team leads ... "they struggled to allocate a budget, or a good enough budget to adequately do user testing". It is both costly and time consuming and clients often do not understand the value of implementing a proper UX process or do not have the resources to fund them. A design team lead mentioned ... "Many clients think that they understand what UX means, but then aren't willing to pay for what UX actually is." Lermen, et al. [37] found that although preincubated startups implement a variety of qualitative UX methods, there is often a lack of quantitative feedback from users. They found that the higher the maturity of the organisation the higher the need for more agile UX practises are required to collect feedback form a user base [37].

Interviewees found the facilitation of communication between different stakeholders due to client availability challenging. The agile manifesto promotes customer collaboration over contract negotiations since it ensures the development team is aligned with the customer's needs, but this is difficult to facilitate when clients are not available to provide feedback. Clients often are unsure of what they need and do not have a clear idea of the requirements of their product [38]. Clients may provide inadequate requirements to the business analysts or designers which can remain a problem all the way through to the development stage if regular alignment meetings are not held with the relevant stakeholders. Regular alignment meetings would mediate the challenges of communicating project constraints and help to manage the clients' expectations.

One agile core principle state that “Working software over comprehensive documentation” while some participants stated that the management of knowledge through effective API documentation and light requirements documentation mitigates the challenge of being dependant on individuals for project progress. Requirements tend to get miscommunicated if there are several points people between the customer and the developer, and with that Agile promotes light documentation, many teams tend to forgo the proper documentation of requirements for verbal communication. According to Alvarez, et al. [38] it is essential that requirements are properly documented in order to anticipate and control any changes in the requirements that may arise during the course of the project.

During large projects it is often difficult to get representative form each stakeholder group as a dedicated member of the project team and often with government related projects interactions with the client are few and far between. This issue can be intensified by the proximity of the client to the software team as it is not always feasible to visit the client on sight to observe the end user or to demo to the client. According to Zamani and Pouloudi [39], distance inhibits shared understanding between members of a team and between a client and the software provider. Proximity enables more frequent interactions and allows for more effective communication [39].

Project constraints. The biggest challenge related to project constraints challenges with project budget. Often provision is not made in the budget to include design processes, or a project has a fixed cost and thus has limited allocated time and resources to create user centred products. In cases where organisations have insufficient budget for extensive and in depth design processes, literature suggest the use of agile UX [37]. Agile UX implement a combination of design thinking, lean startup and lean UX to encourage innovation and delivering value to users [37]. Another challenge that was discussed related to the availability of testers. In SME’s there are often a limited number of testers employed in the organisation and their time needs to be distributed between all the active projects. Because of the fast pace of agile it is often difficult for these testers to adequately test the product before the next phase of the project is initiated. Saeeda, et al. [40] found that the four main challenges faced with testing in large scale agile projects were a lack of system testing, a lack of acceptance testing, a lack of regression testing and a lack of integration testing. Because clients value time to market the team often is not given the opportunity to pilot test the product before it is released into production. Since the business analysis and design phase of the SDLC is often short this may cause edge cases that were not catered for during development to become prevalent only once the product is being implemented. Clients often do not do user acceptance testing due to a restriction on resources [40].

Challenges faced by developers. Developers indicated that time and budget constraints were to major factors that impact their work. This correlates with a model known as the agile triangle. The three dimensions of the agile triangle consists of value, quality and constrains [41]. This fell into the category of project constraints. According to Kothuru Chinnadurai [41] value should remain a fixed goal while the

constraints are changed throughout the project life cycle to maximise the value output. The budget constraints the developers experienced often affected the amount of time they were allowed to spend on certain features. This correlates with findings by Saeeda, et al. [40] finding that due to strict deadlines and a lack of resources developers often skip important processes to stick to project timelines. This could be due to the project being a fixed cost project or the client not buying in to the implementation of user centred design processes. This interlinks with the time constraints experienced by developers, as often spending time to improve a feature for the sake of the user experience was not feasible due to timelines and budget constraints.

Process Challenges. The biggest process challenge faced by interviewees related to the fast pace of agile. Designers found it difficult to effectively empathise with the user, define the needs and requirements, ideate on a variation of potential solutions, create designs and working prototypes as well as validating their assumptions through user testing while following the agile method. Designers also found it difficult to communicate the bigger picture of the project to developers if the project had not yet been effectively scoped out and analysed. Persson, et al. [4] explained on how UX design and software development processes can be adjusted to maintain quality and agility, a UX designer indicated that it is extremely difficult to create wireframes once the development phase has begun, as analytics and due diligence disappears, and the design team loses track and overview of the work that they are doing. Thus, the fast pace at which agile requires sprints to be completed is not an effective project management method for the design phase of a project.

Design Challenges. There are often requirement changes at a late stage of the SDLC, making it difficult for the project to remain user centred while meeting timelines. Developers may also sometimes implement changes to the designs without consulting the designers once requirement changes are requested. This does not adhere to the agile manifesto value stating “Respond to changes rather than following a plan” however this makes it difficult to ensure the minimum level of user experience is maintained throughout. Persson, et al. [4] suggests that designers need to take on an active role of reviewing the implementation of their designs during the development phase.

Agile Principles that impede empathic design. The biggest principle of agile that impedes empathic design, as mentioned by 12 of the interviewees, is the speed at which software needs to be developed as it does not allow time to complete rigorous research. Speed is also an inhibitor of creative design thinking, as a design lead indicated "I found designers often a bit slower than dev in terms of when the actual agile side of things". According to a study done by Acar, et al. [42] constraints can benefit creativity as proven by numerous startups where input constraints (time, money etc), output constraints (customer feedback) and process constraints (daily stand-up's) can increase creative performance but that the creative process faces challenges after a constraint threshold has been reached. The goal is thus to identify the optimal level of constraints that will not inhibit the creative process and allow for innovation [42].

The interviewees commented that the design processes followed by Organisation X followed more of a Waterfall approach during the design phase. Meaning the entire project is scoped out and designed before moving on to the development phase. According to Stepanova [9] this is known as the “design-agency” model and describes this model as ineffective due to the heavy documentation handover. This can be mitigated by implementing a hybrid agile approach that allows the design team to conduct the design phase of the SDLC following the waterfall method, then switching over to agile once the development phase is reached.

The interviewees argue that to properly address the problem, the entire solution must be scoped out to cater for all situations and user stories. This allows them to have a holistic view of the problem [9]. Understanding the entire system allows the designers and business analysts to better fulfil their role as developer support during the development phase. The design team has overcome the “heavy documentation handover” by breaking the designs up into “Dev packages” that consists of enough work to fill a sprint, complete with handover notes and requirement explanations. This corresponds with the regular alignment meetings recommended by literature that allows for better quality UI development [9, 33, 34]. It is also recommended that designers should regularly have stand-ups, retrospectives, and sprint planning sessions with the development team as well as ad-hoc meetings to discuss any shortcomings that was noticed in the designs [9, 33, 34].

Recommendations. After an analysis of the themes that were prominently mentioned in the interviews the following recommendations can be made:

- If an Agile software team is experiencing a lack of client buy in with regards to UX design, it is recommended that the team take on a Lean UX approach. This way the team can maintain a minimal level of usability in the product that they are creating.
- Make use of heuristic reviews and rely on the design team’s expert knowledge of the design principles if there are strict budget constraints such as a fixed cost project or if it is difficult to find users to perform user tests.
- If it is difficult to find expected end user to perform user tests with, use readily available people in the team’s vicinity. This includes friends and family, as well as colleges. It is better to test with some users than to not test at all. The data gathered from such tests must be analysed with the knowledge that the tester sample does not perfectly match the target user group.
- Maintain light documentation to minimise the dependency on a small group of individuals who possess a large portion of the knowledge on a project. Team members may get sick or leave the company, or the team might expand, in which case new team members need to be onboarded and it is thus important to have a single source of truth from which to reference.
- Create sustainable API documentations and useful code comments. This allows new members to be onboarded with ease.
- Designers should create hand over documents for developers that are easy to consume. These “Dev Packages” should contain in depth information on the features

that were selected to be worked on in the upcoming sprint. The documentation should be maintained if any changes are made during the sprint.

Critical Success Factors. The biggest success factor that was identified during the interview process is the flexibility of Organisation X in the application of its agile principles to fit with the needs of their clients. This Hybrid Agile project management approach allows them to incorporate the strengths of both agile and waterfall into their projects. According to Adelakun, et al. [43] some of the advantages of Hybrid Agile include adaptability, developer happiness, increased product quality and early defect detection. Adelakun, et al. [43] also emphasises the usage of Hybrid Agile based on client preference.

Similar to the findings in our interviews, Adelakun, et al. [43] advocates strongly for adequate documentation to be done. While literature encourages minimal documentation [44-46], the design team at Organisation X found that light documentation is needed to avoid dependency on single individuals that are the sole owner of knowledge. The design team make developer handover files with in-depth explanations next to the high-fidelity designs to minimise the locations in which developers need to go searching for information. The development team documents all endpoints that have been created so that sustainable development can take place.

It is critical for the implementation of empathic design that all members of the software development team imitate the user at some level. This enables the team members to take on more of an observational role before they switch to the problem-solving role. This allows for the creation of more pleasurable and impactful products [10].

Organisation X implements Lean UX in low budget fixed cost projects, maintains a minimal level of usability through the practise of heuristic evaluations and following design principles. This is also Implemented when a project lacks the buy in from a customer to implement the full UX design process. According to Alhammad and Moreno [47] Lean UX has a lightweight nature that follows iterative processes based on design thinking, which allows it to fit well with Agile projects. Some of the challenges faced by Alhammad and Moreno [47] included that their research subjects found it difficult to fit Lean UX processes to fit into the short sprints required by the agile methodology, which aligns with the findings form the interviews conducted for this study.

5 Conclusion

The flexibility that organisations allow for in their agile processes can be a great benefit with regards to effectively matching the needs of their clients. The implementation of Hybrid Agile can increase the quality of the final product, increase the adaptability of the team, and increase the overall happiness of the team. The flexibility allows for more time to manage the knowledge regarding the project as well as increasing the allocated time for analysis and design. Teams can increase their empathy for their end users by imitating the user at each state of the product development cycle,

sharing the responsibility of user centred thinking. It is essential to maintain a minimum allowable amount of usability of the product by performing Lean UX practises, even if the project has a small design budget. Testing the product is critical to ensure the product is fit for purpose, usable and delightful.

6 References

1. S. Al-Saqqa, S. Sawalha, and H. AbdelNabi, "Agile software development: Methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, 2020.
2. C. Ekman and J. Martinsson, "Adapting Rapid Contextual Design for Smartphone App Development: User-Centered Design for Small Teams," *Teknisk- naturvetenskaplig fakultet Uppsala Universitet, Uppsala, UPTEC IT 14 015*, 2014.
3. D. Sy, "Adapting usability investigations for agile user-centered design," *Journal of usability Studies*, vol. 2, no. 3, pp. 112-132, 2007.
4. J. S. Persson, A. Bruun, M. K. Lárusdóttir, and P. A. Nielsen, "Agile software development and UX design: A case study of integration by mutual adjustment," *Information and Software Technology*, vol. 152, p. 107059, 2022.
5. H. Obendorf and M. Finck, "Scenario-based usability engineering techniques in agile development processes," in *CHI'08 extended abstracts on Human factors in computing systems*, 2008, pp. 2159-2166.
6. R. Hoda, J. Noble, and S. Marshall, "Agile undercover: when customers don't collaborate," in *International Conference on Agile Software Development*, 2010: Springer, pp. 73-87.
7. T. Memmel, F. Gundelsweiler, and H. Reiterer, "Agile human-centered software engineering," in *BCS-HCI'07: 21st British HCI Group Annual Conference on People and Computers*, Lancaster, UK, 2007, vol. 1: ACM, pp. 167-175.
8. M. Detweiler, "Managing UCD within agile projects," *interactions*, vol. 14, no. 3, pp. 40-42, 2007.
9. V. Stepanova, "Aligning UX designers and UI developers work with design systems," 2021.
10. A. Devecchi and L. Guerrini, "Empathy and Design. A new perspective," *The Design Journal*, vol. 20, 1, pp. S4357-S4364, 2017.
11. E. Köppen and C. Meinel, "Empathy via design thinking: creation of sense and knowledge," in *Design thinking research*: Springer, 2015, pp. 15-28.
12. A. Heylighen and A. Dong, "To empathise or not to empathise? Empathy and its limits in design," *Design Studies*, vol. 65, pp. 107-124, 2019.
13. A. Gasparini, "Perspective and use of empathy in design thinking," in *ACHI, the eight international conference on advances in computer-human interactions*, 2015, pp. 49-54.
14. R. Mokhtar and M. Khayyat, "A Comparative Case Study of Waterfall and Agile Management," *SAR Journal (2619-9955)*, vol. 5, no. 1, 2022.
15. N. Dwivedi, D. Katiyar, and G. Goel, "A Comparative Study of Various Software Development Life Cycle (SDLC) Models," *International Journal of Research in Engineering, Science and Management*, vol. 5, no. 3, pp. 141-144, 2022.
16. A. Mishra and Y. I. Alzoubi, "Structured software development versus agile software development: a comparative analysis," *International Journal of System Assurance Engineering and Management*, pp. 1-19, 2023.

17. J. Patton, "Hitting the target: adding interaction design to agile software development," in *OOPSLA 2002 Practitioners Reports*, 2002, pp. 1-ff.
18. 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*, p. 106957, 2022.
19. D. Strode, T. Dingsøy, and Y. Lindsjorn, "A teamwork effectiveness model for agile software development," *Empirical Software Engineering*, vol. 27, no. 2, pp. 1-50, 2022.
20. J. Armitage, "Are agile methods good for design?," *interactions*, vol. 11, no. 1, pp. 14-23, 2004.
21. N. Yahya and S. S. Maidin, "The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team," in *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, 2022: IEEE, pp. 1-5.
22. A. Wale-Kolade, P. A. Nielsen, and T. Päivärinta, "Integrating usability practices into agile development: A case study," in *International Conference on Information Systems Development: 02/09/2014-04/09/2014*, 2014: University of Zagreb, pp. 337-347.
23. M. A. S. Al Jabri, "Enhancing Products Delivery Through the Application of Innovative Operating Model Based on Hybrid Agile Delivery Method: Case Information Communication Technologies "ICT" Service Providers," in *BUID Doctoral Research Conference 2022: Multidisciplinary Studies*, 2023: Springer, pp. 35-46.
24. N. Yahya, S. S. Maidin, and M. S. Safari, "The Development of Interview Protocol to Explore Hybrid Agile Software Development Phases," *International Journal*, vol. 10, no. 3, 2021.
25. C. Putnam, E. Rose, E. J. Johnson, and B. Kolko, "Adapting user-centered design methods to design for diverse populations," *Information Technologies & International Development*, vol. 5, no. 4, pp. 51-74, 2009.
26. M. Zorzetti, I. Signoretti, L. Salerno, S. Marczak, and R. Bastos, "Improving Agile Software Development using User-Centered Design and Lean Startup," *Information and Software Technology*, vol. 141, p. 106718, 2022.
27. T. M. Ponciano, C. Campese, and J. Mascarenhas, "Concept test method to assist engineering students to involve users in a user-centered design project," *Product: Management and Development*, vol. 19, no. 1, pp. 0-0, 2021.
28. A. Jones, V. Thoma, and G. Newell, "Collaboration constraints for designers and developers in an agile environment," in *Proceedings of the 30th International BCS Human Computer Interaction Conference 30*, Bournemouth, UK 2016: BCS Learning and Development Ltd., pp. 1-9.
29. Oxford dictionary of English, *Oxford dictionary of English*. Oxford University Press, USA, 2010.
30. M. Kouprie and F. S. Visser, "A framework for empathy in design: stepping into and out of the user's life," *Journal of Engineering Design*, vol. 20, no. 5, pp. 437-448, 2009.
31. D. K. Ho and Y. C. Lee, "The quality of design participation: Intersubjectivity in design practice," *International Journal of Design*, vol. 6, no. 1, 2012.
32. K. D. Elsbach and I. Stigliani, "Design thinking and organizational culture: A review and framework for future research," *Journal of Management*, vol. 44, no. 6, pp. 2274-2306, 2018.
33. T. Sedano, P. Ralph, and C. Péraire, "Dual-Track Development," *IEEE Softw.*, vol. 37, no. 6, pp. 58-64, 2020.
34. M. Meingast, T. Ballew, R. Edwards, E. Nordquist, C. Sader, and D. Smith, "Agile and UX: The Road to Integration The Challenges of the UX Practitioner in an Agile

- Environment," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 1002-1006, 2013, doi: 10.1177/1541931213571224.
35. O. Mazni, S.-A. Sharifah-Lailee, and Y. Azman, "Agile documents: Toward successful creation of effective documentation," in *International Conference on Agile Software Development*, Norway, 2010, vol. 48: Springer, pp. 196-201.
 36. I. Bongiovanni and C. P. Louis, "Theory and practice of Design Thinking: perspectives of designers and business consultants," *International Journal of Design Creativity and Innovation*, vol. 9, no. 3, pp. 174-191, 2021.
 37. F. H. Lermen, P. K. de Moura, V. B. Bertoni, P. Graciano, and G. L. Tortorella, "Does maturity level influence the use of Agile UX methods by digital startups? Evaluating design thinking, lean startup, and lean user experience," *Information and Software Technology*, vol. 154, p. 107107, 2023.
 38. S. Alvarez, K. Duy, M. Zapata, J. Galarza, D. Martinez, and C. Puco, "The communication between client-developer in the process of requirements elicitation for a software project," in *Trends and Applications in Information Systems and Technologies: Volume 4 9*, 2021: Springer, pp. 36-45.
 39. E. D. Zamani and N. Pouloudi, "Shared mental models and perceived proximity: a comparative case study," *Information Technology & People*, vol. 35, no. 2, pp. 723-749, 2022.
 40. H. Saeeda, M. O. Ahmad, and T. Gustavsson, "Challenges in Large-Scale Agile Software Development Projects," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023, pp. 1030-1037.
 41. B. Kothuru Chinnadurai, "Analyzing Agile Project Success Criteria from a Practitioner Perspective," ed, 2020.
 42. O. A. Acar, M. Tarakci, and D. Van Knippenberg, "Creativity and innovation under constraints: A cross-disciplinary integrative review," *Journal of Management*, vol. 45, no. 1, pp. 96-121, 2019.
 43. O. Adalakun, R. Garcia, T. Tabaka, and R. Ismail, "Hybrid project management: Agile with discipline," 2017.
 44. I. Hadar, S. Sherman, E. Hadar, and J. J. Harrison, "Less is more: Architecture documentation for agile development," in *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2013: IEEE, pp. 121-124.
 45. W. Behutiye, P. Rodríguez, M. Oivo, S. Aaramaa, J. Partanen, and A. Abhervé, "Towards optimal quality requirement documentation in agile software development: a multiple case study," *Journal of Systems and Software*, vol. 183, p. 111112, 2022.
 46. D. D. Gregorio, "How the Business Analyst supports and encourages collaboration on agile projects," in *2012 IEEE International Systems Conference SysCon 2012*, 2012: IEEE, pp. 1-4.
 47. M. M. Alhammad and A. M. Moreno, "Integrating user experience into Agile: an experience report on lean UX and Scrum," in *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*, 2022, pp. 146-157.