ISSN : 2350-0743

International Journal of Recent Advances in Multidisciplinary Research

Vol. 11, Issue 05, pp.9888-9892, May, 2024

RESEARCH ARTICLE

USER-CENTERED DESIGN IN AGILE: INTEGRATING PSYCHOLOGICAL PRINCIPLES FOR ENHANCED USER EXPERIENCE

***Hilal Durak, Harun Gultekin, Dimitrios Haralampopoulos, Ioannis Skoulidas, Nicholas Ciraulo, Pavan Kumar Vipparla and Sarah Korczukowski**

USA

ARTICLE INFO

Article History

Received 08th February, 2024
Received in revised form
20th March, 2024
Accepted 27th April, 2024
Published online 30th May, 2024

Keywords:

Considerations,
Prioritized, Throughout,
Ensuring.

ABSTRACT

In Agile product development, user consideration is fundamental to success. Users serve as the nucleus of a product, rendering developers without a purpose in their absence. Consequently, User-Centered Design (UCD) emerges as a pivotal strategy within Agile Development, emphasizing user needs throughout the process. This paper delves into the myriad applications of psychological principles in UCD, elucidating how stakeholders at every stage of the Agile Development Life Cycle stand to gain significant advantages.

Citation: Hilal Durak, Harun Gultekin, Dimitrios Haralampopoulos, Ioannis Skoulidas, Nicholas Ciraulo, Pavan Kumar Vipparla, Sarah Korczukowski. 2024. *"User-Centered Design in Agile: Integrating Psychological Principles for Enhanced User Experience", International Journal of Recent Advances in Multidisciplinary Research,* 11, (05), 9888-9892.

INTRODUCTION

In the landscape of contemporary software development, the fusion of Agile methodologies with User-Centered Design (UCD) principles emerges as a focal point of both academic inquiry and practical application. The manner in which users interact with software products is intricately influenced by their cognitive capacities, social behaviors, and individual preferences (Adhikari). Teams dedicated to UCD principles in software development prioritize user experience across the entirety of the development lifecycle. This literature review undertakes a critical examination of how psychological principles are integrated into UCD within Agile frameworks, with a focus on understanding their impact on user engagement and satisfaction. Central to this investigation is the recognition of users not merely as end-consumers, but as indispensable contributors to the iterative development process of software. As software engineering adapts to keep stride with technological advancements and escalating user expectations, the amalgamation of Agile's rapid, iterative cycles with UCD's user-centric ethos assumes increasing significance. A broader adoption of UCD, underpinned by a more profound comprehension within the UCD community, is imperative (Mao *et al*., 2001).

This concerted approach aims to craft software solutions that not only meet but exceed user expectations. Drawing insights from 20 scholarly articles, our exploration spans a spectrum of discussions, encompassing strategies for and challenges of integrating UCD into Agile processes (Fokkinga *et al*., 2020). This review navigates through key areas, including the hurdles associated with user involvement and the application of psychological models to refine the UCD process within Agile environments. The perspectives gleaned from these scholarly sources enrich the ongoing academic discourse surrounding the integration of Agile and UCD, offering invaluable insights for software development practices attuned closely to user needs.

METHODS

LITERATURE REVIEW

A comprehensive literature review was conducted to explore the integration of user-centered design (UCD) principles within agile development processes.

Electronic databases including IEEE Xplore, ACM Digital Library, and Google Scholar were systematically searched using keywords related to UCD, agile methods, and software development. The search scope was confined to papers published between 1995 and 2023, ensuring a contemporary understanding of the subject matter. This review aimed to capture a diverse range of perspectives and approaches regarding the integration of UCD within agile methodologies.

Relevant papers were meticulously selected based on their title, abstract, and full text, followed by a thorough screening process to determine eligibility. In addition to seminal works by Mao *et al*. (2001) and Cockton *et al*. (2016), recent contributions such as those by Birgersson (2021), Fokkinga *et al*. (2020), and Malik *et al*. (2021) were also incorporated. These contemporary sources provided valuable insights into emerging trends, challenges, and best practices in the intersection of UCD and agile development. The synthesized findings from the literature review inform the discussion section of this paper. It delves into the common themes identified across the reviewed papers, such as the importance of iterative development, simple representations of design artifacts, and the application of psychological models in UCD. Furthermore, the discussion critically examines the challenges and opportunities associated with integrating UCD principles within agile development processes, drawing on insights from a range of contemporary sources. For instance, Hennel and Rosenkranz (2021) discuss the importance of psychological safety in agile information systems development, shedding light on avenues for improvement beyond traditional UCD approaches.

Data Extraction

Papers meeting the predetermined inclusion criteria underwent meticulous analysis to extract pertinent information regarding the integration of UCD principles and agile methods in software development projects. Data extraction encompassed identification of key themes, challenges, and strategies elucidated by authors, alongside any empirical evidence supporting the effectiveness of integrating UCD with agile approaches. For instance, Adhikari *et al*. (2019) highlighted the importance of user-centricity in Agile development, reinforcing the significance of UCD principles.

Data Synthesis

A thematic analysis approach was employed to synthesize the findings from the selected papers. Themes related to the integration of UCD and agile methods, including iterative/incremental development, representation of design artifacts, and psychological models in UCD, were meticulously identified and categorized. For instance, Fokkinga *et al*. (2020) emphasized the iterative nature of UCD in Agile environments, underscoring its role in refining user experiences iteratively.

Limitations

While this study endeavors to offer comprehensive insights into the integration of user-centered design (UCD) principles within agile development processes, several limitations merit acknowledgment. Primarily, this study is contingent upon the availability and scope of the literature reviewed. Despite exhaustive efforts to identify relevant papers, it remains plausible that some pertinent studies may have been inadvertently overlooked. Moreover, the generalizability of the findings may be confined to the context of software development projects, potentially limiting their applicability across other domains. Notwithstanding these limitations, the findings of this study provide valuable insights into the synergistic integration of UCD principles within agile development methodologies. They underscore the significance of iterative/ incremental development, simplistic representations of design artifacts, and the application of psychological models in UCD for enhancing the usability and user experience of software products. Nevertheless, further research endeavors are warranted to explore the efficacy of specific strategies for integrating UCD and agile methods across diverse contexts.

DISCUSSION

Across the papers reviewed, a common theme was the integration of user-centered design principles in the agile development process to best suit the needs of the end-user throughout the development of software (Brhel *et al*., 2015; Hussain *et al*., 2009; Cockton *et al*.,2016; Gulliksen *et al*., 2010). While multiple papers brought forward valid points about the difficulty with integrating some aspects of the UCD with agile development, such as the need for a usability expert which goes against the team-oriented approach found in agile, all papers acknowledged the congruence with which UCD, and agile development could be applied together. One principle commonly brought up by papers was the need for an iterative/incremental approach to development (Brhel *et al*., 2015, pg. 171-172; Gulliksen *et al*., 2010, pg. 402). This already exists as the basis for development in agile methods, asit "approaches evaluate production-ready code at the end of each iteration," (Salah 2009), however it is also important to the design philosophy of UCD, thus it remains an important part of the integration of UCD and agile methods. For instance, Gulliksen *et al*. emphasizes how a UCD approach requires continuous iterations with the users, as well as incremental updates (2010, pg. 402). This allows users to constantly chime in on the software that is being produced to give their input and keep the development centered around the needs/wants of the user. Keeping the user centered at each step of the development process keeps developers on track to produce software that best suits the user. In agile development, a similar approach is taken, that is using iterative/incremental development to continually reassess/revaluatewhat is being developed, keeping key shareholders centered at the process. While like the UCD approach, the framing of development around shareholders, as opposed to users, means that agile methods will not necessarily lead to usable software, simply software that has a useful value. Thus, we found that many papers emphasized iterative development which includes users at every step, such as having end-users on the development team to give direct feedback and each iteration. Adding to this idea, Brhel *et al*. even suggests having multiple interwoven parallel teams, some working on software development, others solely working on design (2015, pg. 172-173). This includes starting with an initial iteration solely working on the design of the software in question, then moving onto parallel work on development and design.

Having teams which focusing entirely on the design aspects of a project will allow these teams to fully focus on the end-user, while the development teams can be left to focus on software development, looking to the design teams for answers relating to features exposed to the end-user. Another product design principle held to be important at the intersection of UCD and agile methods is the importance of simple representations of the design of the system through updated artifacts. (Brhel *et al.*, 2015; Cockton *et al.*, 2016; Gulliksen *et al.*, 2010).Artifacts, such as burn-down charts, are an already commonly used aspect of agile methods. Having a visual representation of the development process can quickly establish remaining components of a project to be done and can allow developers to make time considerations for specific components of the project. Additionally, artifacts are a quick way to represent the ongoing state of a project to key shareholders, which can establish what work is being done and if parts of the development process must change to accommodate changing requirements for the project. When integrating UCD into this framework, the key point multiple papers made was to keep representations simple and understandable such that an end-user would be able to track the progress of a project and remain informed about the design of the project. If users are not able to fully understand the consequences of the system's design, it is for them to misunderstand what is being developed for them and thus be unimpressed by the final product when they can use it. For instance, Gullkisen *et al.* point out how representations, such as UML prove to be too complicated for the end-user and hide the over-arching design/development from the user (2010, pg. 402). While the abstractions of UML may be conducive to more useful software being developed, it could be at the expense of the usability of the software. Integrating this aspect of UCD with agile methods becomes simple then, since agile already focuses on having simple representations of the development process, and already makes simplifying assumptions, such as the use of user stories.

Another prominent aspect of UCD that is important to consider is the type of framework being used to design the product regarding the users in each step of the UCD process. Birgersson describes the main stages of UCD as follows: Empathize, Define, Ideate, Prototype, and Test. In the Empathize step, the developers research the users' needs. They empathize with the users to understand what kinds of problems they would like to have solved by the piece of software being built. Afterwards, the developers state the users' needs and problems concretely in the Define step. Developers then generate ideas for solutions to the listed problems from the Define step in the Ideation step. After picking through the ideas and generating a solid basis for the product, they prototype it, hence it being called the Prototype step.The prototype is then evaluated thoroughly and tested in the Testing step. This is an iterative process, that can and usually does repeat multiple times before a final product is reached. Birgersson describes various psychological models to increase the effectiveness of the UCD process (Birgersson, 2021). The first of such models is the Nudge Theory model. Nudge Theory is something that most of us are familiar with, even if we do not know the theory by name. It takes advantage of the fact that choices made by individuals can be easily affected in predictable ways without forbidding any options from the user. This type of behavioral suggestion is called a nudge (Birgersson).An example of Nudge Theory is using a larger and more colorful button for the option you would prefer the user to click, while keeping the less desirable choice present but less noticeable. An ideal application for something like Nudge Theory would be like asking a user if they would like to apply recommended security or sustainability features for that product. The option for the developers wants us to click ("Yes, enable these features") would be in a brighter color with a larger button, while the other option(s) ("No, thank you" or "Remind me later") are in a more subdued color scheme and have a smaller button target area (Birgersson, 2021). The second such model discussed is the Hooked Model. Its purpose is to get users to continuously engage in the workflow associated with the product. It is comprised of four steps: Trigger, Action, Variable Reward, and Investment. The cycle starts with the Trigger. This Trigger is usually some need of the user, for example, hunger. The user then takes an Action to satisfy the need brought on by the Trigger, such as visiting an app.A Variable Reward, like rewards points, is then given to the userafter taking the action. It is important for the reward to vary such that the user does not tire of the reward system. Investment in the system by the user is the last step in the cycle. The reward or thing that they sought is saved in some way by the product. This encourages the user to come back to this product or service again to satisfy the need they felt initially in the Trigger step. This completes the loop and primes the user for the next iteration (Birgersson, 2021).

Some fewer tangible models and theories discussed by Birgersson include Mindful Design Theory, Theory of Interpersonal Behavior, and Transtheoretical Model of Behavior Change (TMBC). The most powerful of the three is TMBC, which hassix stages. First is the Pre-Contemplation stage, where the development team works with potential users and helps them evaluate previous choices,they have made about things relating to the product's topic of focus and place them in a new perspective, gearing them for the mindset shift to come. The Contemplation stage follows, where the developersfocus on raising awareness among the users of the importance and benefits of changing their software or habits. The Preparation stage prepares the users, empowering them to create an action plan, which flows directly into the more concrete Action step where thedesign should support the user to take the desired corrective actions. The Maintenance step, like the Variable Reward step in the Hooked Model, exists to reward the user with something to prevent relapsing into undesirable behavior. The Termination step is executed when the user is ready to sustain the new behavior on their own without assistance from the product or development team (Birgersson, 2021). Analyses of these psychological UCD models and theories by Birgersson made through interviews reveal that each of them has optimal times during the UCD lifecycle to be applied. Interviewees felt that Nudge Theory is good for the beginning of the cycle, namely in the Ideate and Prototype stages. They also warn that Nudge Theory may be rendered ineffective or even backfire if there is no user interest to be nudged. The Hooked Model was thought to be useful throughout the process, but most prominent in the Empathize step since it is important for the developers to know what the users need to initiate and satisfy the Trigger. It is also relevant in Ideate, as the developers need to produce a way to trigger the need and reward the Action.

Transtheoretical Model of Behavioral Change seemed useful primarily in the initial stages of development, but interviewees said that it could be useful at every point during the design process and is something good for the developers and designers to have in the back of their minds (Birgersson, 2021). Birgersson ends the paper with a warning from the interviewees regarding the psychological methods reviewed within. The papertackles Sustainable UX:that is, "all aspects of the end-user's interaction with the company, its services, and its products,"designed to not compromise on social or ecological issues in favor of sustainability. Interviewees said that the theories themselves are not sustainable per se — they are only sustainable when that is the designer's intention. All the theories described manipulate the user to produce the desired behavior from them. Supported by Malik (2021), it is important to understand"… the effects of agile practices on the psychological states that may affect the outcomes of" these projects and companies. It is extremely important that this point is emphasized so that the theories are not used maliciously or solely for financial gain (Birgersson, 2021).

With that warning in mind, we must also recognize the other problems that arise when attempting to implement UCD in an Agile environment. There is no "one-process-fits-all" approach to Agile, and UCD is no exception. In Gulliksen *et al.*'s 1999 paper User Centered Design — Problems and Possibilities, some problem areas of UCD were addressed by talking with attendants of a Participatory Design Conference in Seattle that year. The major concerns that were brought up by these attendants were User Participation, Project Management & Work, Organization, and Communication. They were able to gather feedback from users early and frequently, ensuring that the software evolved based on the actual users' experiences. How do we get users to participate actively in the design process? How do we make sure that potential users feel like their input matters? How do we structure the project hierarchy? How do we make sure that everyone is on the same page (Adhikari)?.

For starters, it isof utmost importance to the UCD process to find out who the users are andtalk to them! As stated by Garcia, Silva and Silveria(2017), "communication is crucial for the success of any business". Even if the product is niche and developers can only find a handful of potential users, that is better than none (Gulliksen, 1999). In fact, Kujala and Kauppinen (2004)discussed a few methods to identify and select users for UCD.They define what they call a Persona, a "precise description of a hypothetical user and [their] goals," and it represents a group of users throughout the entire design process. Schneiderman *et al.* (1998) states that all design should begin with an understanding of the intended users, including population profiles that reflect age, gender, physical abilities, education, cultural or ethnic background, training, motivation, goals, personality and more.This information can be collected by talking to potential users and results in the formulation of a comprehensive user profile or Persona. Sometimes developers may have a vague or contradictory sense of their user base, especially if there are many heterogeneous users. Knowing that according to the human psychology, human beings create their own world and have different worlds from others (Cömertpay& Durak, 2023). Cooper's (1999) solution to this issue is to narrow the design target to a particular group of users, even if it represents a minority of them. Cooper shows that this method can be successful as well, especially for products such as automobiles or other products for which the number of customers and users is extremely high.Hackos and Reddish (1998)formalized a process for user identification and selection of the following structure: Brainstorm a preliminary list of users, describe the main user characteristics, followed by the user groups, and order them by priority, select typical and representative users from the groups, and finally gather information from the users and redesign the groups. This emphasizes the iterative nature of user identification, and furthermore shows that personas and user profiles should be based on real data collected from real people to enhance the credibility of the models. Addressing the other issues raised by Gulliksen (1999), it is important that before a company engages in UCD,there is a Mutual Shared Understanding between the developers, users, and upper management. For example, IT people sometimes make the mistake of propagating the attitude that "if you just give us a requirements specification, we can create a system that will suit your needs". This creates expectations that cannot be met. The users also must understand the limitations of what the developers can do within reason.That is why it is important to be able to involve the users in the UCD process as much as possible, such that Mutual Shared Understanding is strengthened. It is also important that with that Mutual Shared Understanding comes collaboration and equality. Gulliksen states that it is important to "let the users leave their trace in the design office" and have them feel like their thoughts and opinions will have real bearing on the product.

CONCLUSION

In conclusion, this literature review has critically explored the integration of User-Centered Design (UCD) principles with Agile methodologies in the context of software development. By examining 20 scholarly articles, this analysis brought to light the essential consideration of end-user needs throughout the Agile Development Life Cycle, with a particular focus on the contribution of UCD to the development of products that adequately meet these needs. Our findings algin with the work of Hennel and Rosenkranz(2021), who emphasized the importance of investigating socio-technical development, highlighting the case for psychological safety in agile information systems development. The consensus across the reviewed literature points to a balanced interplay between UCD and Agile development practices. This balance is primarily achieved through an iterative and incremental approach that is central to both methodologies, promoting ongoing engagement with end users. This is supported by Laanpere(2018) having stated, "Efficient communication between the agile team members and the UX designers is of utmost importance for the integration of UCD approach and agile methods". The literature suggests the formation of parallel teams concentrating on design aspects, thereby ensuring that user considerations are prioritized throughout the development process. It also recommends the use of visual tools, such as burn-down charts, which are congruent with UCD's advocacy for simplicity and effective communication of project status. Furthermore, the papers explore the role of psychological models, such as Nudge Theory and the Hooked Model, in optimizing the UCD process.

These models offer strategies for improving user engagement, satisfaction, and behavior, reinforcing the iterative nature of UCD and its potential to enhance user involvement and feedback mechanisms. However, the literature review also identifies challenges associated with implementing UCD within an Agile framework. These challenges include difficulties in user participation, project management, organizational structure, and communication. The importance of actively engaging with users, employing personas to inform the design process, and fostering a mutual understanding among developers, users, and management is emphasized as critical to overcoming these obstacles. Roth, Patterson, and Mumaw's (2000) cognitive engineering perspectives further point out the critical issues in UCD and help provide an understanding that there needs to be a balance between UCD and Agile practices. Overall, this review has articulated the nuanced and valuable interrelation between UCD and Agile methodologies in creating software products centered on user needs. As shown through the articles, "there is a clear need for more UX studies integrating with agile development. Therefore, we conclude that there is a need to study more to understand the best working practices of Agile UX," (Beux, J., Bellei, E., Brock, L., Bertoletti, A. C., &Hölbig, C. (2018)) and this was a general theme amongst all the literature reviews. Through this academic inquiry, we contribute to the scholarly dialogue on the integration of UCD principles within Agile environments, providing insights and considerations for researchers and practitioners in software development.

REFERENCES

Adhikari, A. Academics. I. T., & Kathmandu, N. (Year). Human-Centered Design Approaches in Software Engineering: Enhancing User Experience and Productivity. [Include any additional information about the source, such as volume, issue, page numbers, or DOI if available.]

Beux, J., Bellei, E., Brock, L., Bertoletti, A. C., &Hölbig, C. (2018). Agile design process with user-centered design and user experience in web interfaces: a systematic literature review. Latin-American Journal of Computing, 5(2), 53-60.

Birgersson, F. (2021). Sustainable UX: Integrating behavioral psychology in the user-centered design process. 21st Student Conference in Interaction Technology and Design, 6, 134–138.

Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. Information and Software Technology, 61, 163-181.

Cockton, G., Lárusdóttir, M., Gregory, P., &Cajander, Å. (2016). Integrating user-centred design in agile development (pp. 1-46). Springer International Publishing.

Cooper, A. (1999). The inmates are running the asylum. Vieweg+ Teubner Verlag.

Cömertpay, E. & Durak, H. (2024). Examining The Regulatory Role of Age In The Relationship Between Existential Anxiety And Secondary traumatic Stress In Teachers, International Journal of Education Technology and Scientific Researches, 9(26), 53-83.

DOI: ttp://dx.doi.org/10.35826/ijetsar.723

Fokkinga, S., Desmet, P., & Hekkert, P. (2020). Impact-Centered Design: Introducing an Integrated Framework of the Psychological and Behavioral Effects of Design. International Journal of Design, 14(3). Retrieved from https://www.ijdesign.org/index.php/IJDesign/article/view/3869

Garcia, A., Silva da Silva, T., & Selbach Silveira, M. (2017). Artifacts for agile user-centered design: a systematic mapping.

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., &Cajander, Å. (2003). Key principles for user-centred systems design. Behaviour and Information Technology, 22(6), 397-409.

Gulliksen, J., Lantz, A., &Boivie, I. (1999). User centered design—problems and possibilities. ACM SIGCHI Bulletin, 31(2), 25–35. https://doi.org/10. 1145/329 657.32 96 66

Hackos, J. T., & Redish, J. (1998). User and task analysis for interface design (Vol. 1). New York: Wiley.

Hennel, P., & Rosenkranz, C. (2021). Investigating the "socio" in socio-technical development: The case for psychological safety in agile information systems development. Project Management Journal, 52(1), 11-30.

Hussain, Z., Slany, W., & Holzinger, A. (2009). Current state of agile user-centered design: A survey. In HCI and Usability for e-Inclusion: 5th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2009, Linz, Austria, November 9-10, 2009, Proceedings 5 (pp. 416-427). Springer Berlin Heidelberg.

Kujala, S., & Kauppinen, M. (2004). Identifying and selecting users for user-centered design. Proceedings of the Third Nordic Conference on Human-Computer Interaction. https://doi.org/10.1145/1028014.1028060

Laanpere, J. (2018). Revealing the secret pieces of the puzzle: Studying the hidden influences of agile development context on user-centered design.

Malik, M., Sarwar, S., & Orr, S. (2021). Agile practices and performance: Examining the role of psychological empowerment. International Journal of Project Management, 39(1), 10-20.

Mao, J. Y., Vredenburg, K., Smith, P. W., & Carey, T. (2001, November). User-centered design methods in practice: a survey of the state of the art. In Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research (p. 12).

Roth, E. M., Patterson, E. S., & Mumaw, R. J. (2000). Cognitive Engineering: Issues in User-Centered System Design.

Salah, D., Petrie, H., & Paige, R. F. (2009). Towards a framework for integrating user-centered design and agile software development processes. Proc. Irish CHI (Center for Healthcare Innovation) 2009.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., &Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson

****