Organizational Integration of User Experience Contributors in Agile Software Development

Dominik Oelfin

Pforzheim University
Tiefenbronner Straße 65
75175 Pforzheim
E-Mail: oelfindo@hs-pforzheim.de

Frank Morelli

Pforzheim University
Tiefenbronner Straße 65
75175 Pforzheim
E-Mail: frank.morelli@hs-pforzheim.de

ABSTRACT

In today's software industry, agile software development methods have largely replaced traditional processes. At the same time, organizations are increasingly recognizing that user experience plays an important role in the quality and success of software. While agile methods are widely used in practice and therefore influence organizational considerations, the integration of user experience contributors in such organizations poses several challenges. The presented research shows that the involvement of user experience in software development is considered valuable in theory, but its prioritization lags behind in practice. Therefore, we investigate factors that need to be considered to improve the organizational integration of user experience contributors in agile software development. These factors can be divided into structural aspects and aspects of processes and practices. Organizational design, roles involved, and further structural aspects such as organizational autonomy serve as a basis to allow for effective and efficient involvement of user experience contributors in product development processes. Organizational practices shape the way of how the processes of user experience and agile software development can be combined. In particular, the interconnectedness of processes, mutual prioritization, and close collaboration can be identified as key factors. Finally, we consolidate these factors into a framework and enrich them with practical implications. This framework can serve as a starting point for researchers and practitioners to further explore and improve the organizational integration of user experience contributors in agile software development.

KEYWORDS

Agile Software Development, Agile Methodology, Human Centered Design, User Centered Design, User Experience

INTRODUCTION

User experience (UX) has a significant impact on the perceived quality of software, encompassing the "emotions, beliefs, preferences, perceptions, comfort, behaviors, and accomplishments that occur before, during and after use" of software (ISO 9241-210:2019 2019). Roles that are explicitly working on UX are therefore an integral part of organizations for the successful development of software (Bruun et al. 2018, 1–3). As such, the tasks of UX contributors include conducting user research, creating product wireframes and prototypes, assessing usability and accessibility of products and analyzing product interactions (Bruun et al. 2018, 4). Responsibilities often overlap among roles (e.g. "user researcher", "visual designer", "interaction designer" and "information architect"), with some teams combining these distinct responsibilities into a unified UX designer position. The specific naming, responsibilities, and tasks of these roles can vary depending on the organization, product, and context.

The responsibility of UX managers is to plan, oversee, and measure the overall UX process and practice (Bruun et al. 2018, 4). Furthermore, another key objective of UX management is to raise awareness about the importance of UX and the value it creates within the organization (Szóstek 2012). To integrate the UX management strategy, a UX mission should be derived directly from the business goals of the organization (Rosenberg 2019, 30).

The present state of UX influence in an organization can be referred to as UX maturity (Chapman and Plewes 2014, 14). UX maturity has two direct effects on the organization: The more mature the integration of UX into development, the more influence UX has on product decisions. In addition, a higher level of maturity leads to a higher awareness of the importance of UX and thus to a better allocation of resources for the UX team.

In recent years, agile methods have significantly changed the existing organization of software development in many companies (Gerster, Dremel, and Prashant 2018, 1). Teams are becoming more interdisciplinary, development cycles are becoming shorter, and products are brought to market faster (Dima and Maassen 2018, 316). These changes have implications for the integration of UX management into the development process. Where development cycles are more tightly timed, the need for time-consuming UX practices is questioned – the influence of UX contributors seems to fade (McInerney and Maurer 2005, 19). Nevertheless, especially in the context of innovative agile development, UX is a crucial need to develop competitive user-centered products. Interdisciplinary collaboration is becoming increasingly important. The quality of developed software is strongly dependent on the quality of the collaboration of the teams involved and thus significantly determined by organizational aspects.

Studies have empirically demonstrated the influence of organizational structures of software development teams on software quality (Nagappan, Murphy, and Basili 2008). However, their conclusions do not explicitly refer to agile software development and the involvement of UX contributors. Furthermore, various studies on the integration of UX practices and principles in development organizations can be found (Kashfi, Feldt, and Nilsson 2019; Gray, Toombs, and Gross 2015). For the most part, existing publications examine new practices to bring agile methods to UX. However, the necessary organizational approaches to combine UX contributors and development in one organization are only marginally considered yet (Ferreira, Sharp, and Robinson 2012).

Based on this, the following research objectives are covered within this paper:

- What is state of the art for integrating UX in agile software development?
- Which organizational factors influence the integration of UX contributors in agile software development?
- Which implications for the integration of UX contributors in agile software development can be derived from these factors?

We base our research on the Design Science Research method (Hevner et al. 2004; Hevner and Chatterjee 2010). In this paper we present the outcomes of a review of existing literature to provide an overview of the persisting knowledge base. We focus on the organizational design of agile software development teams as well as specifics of structures for UX contributors. Furthermore, we reflect upon existing processes and practices integrating UX practices into agile software development. Hereby we aim to identify influential factors that affect the organizational integration of UX contributors in agile software development and their respective impact on the integration. To add practical insights, enriching the theories gained from literature research, we present a case study that we conducted, analyzing a real-world example of an integration of UX contributors into a specific agile software development organization. Based on the findings from the literature review and the case study we introduce a combined framework containing influential factors and their implications for the organizational integration of UX contributors in agile software development.

ORGANIZATIONAL DESIGN OF AGILE SOFTWARE DEVELOPMENT AND UX

A central concept of agile methods is that their processes benefit from the unique skillsets of each team member and each team (Cockburn and Highsmith 2001, 132). Consequently, based on their strengths, the team members are assigned to specific roles within the agile software development team. Gerster et al. (2020) identify three groups of roles within a product development team: key roles, utility roles and contributor roles. Based on the Scrum methodology, the key roles include product owner, technical product owner and Scrum master (Gerster et al. 2020, 88). Utility roles are fulfilled by the regular team members. Based on their individual skills they are assigned to design, develop, test, integrate, maintain, or operate the product. In addition to this core team, there is an extended team consisting of specialists for specific tasks, brought in as required. These can participate within the team either for short terms but full-time or flexibly as part-time support. Although they enhance the ability of the team it is not intended that they carry out all the tasks related to their field of expertise, but rather transfer their skills to the core team members and support them in their decisions (Gerster et al. 2020, 88).

Gerster et al. (2020) furthermore describe a generic model of an organizational structure for agile software development – the “fully agile unit”. This model is based on an organizational structure used at Spotify, firstly described by Kniberg and Ivarsson (2012) and extended by Gonçalves and Lopes (2014) (Gonçalves and Lopes 2014). The model of a fully agile unit is characterized by the definition of four distinct groupings: tribe, squad, chapter, and guild. The whole of a unit is called a tribe and reflects a product area within a software development organization. Organizations can consist of multiple tribes. Tribes (product areas) consist of multiple products, each developed by one or multiple teams, referred to as squads. Squads are structured like Scrum teams and contain the roles previously described (Kniberg and Ivarsson 2012, 1–5). Several team members of different squads can come together to form a chapter. A chapter typically combines members with similar skills and tasks. Like a matrix structure, each chapter typically has a chapter lead, responsible for all chapter members, who is also part of a specific squad. Alternatively, several team members of multiple squads can form a guild. These are less formal and usually do not have a strict leading role, making them a “community of interest” (Kniberg and Ivarsson 2012, 10). Chapters and guilds are beneficial to keep the separate squads engaged with each other, resulting in economies of scale through reduced work-redundancy, while keeping them as autonomous as possible. Gerster et al. (2020) furthermore add shared teams to the existing concepts.

Involvement of UX

Gerster et al. (2020) position UX contributors as part of the extended team, being assigned full-time while actively participating for short terms. Apart from this generalized positioning, Gerster et al. (2020) do not elaborate further on the integration of UX contributors in the fully agile unit. Kniberg and Ivarsson (2012) do not explicitly mention UX. However, a screenshot in their article shows that UX could be considered as a separate squad. We assume that the concept of shared teams is relevant with regards to the integration of UX contributors into the organizational model. Furthermore, positioning UX contributors as part of the core team can be envisioned as well. To further analyze the positioning of UX, we now focus on the organizational structures that UX contributors might find themselves in.

Organizational Structures for UX Contributors

Centralized Structure

A centralized UX structure is characterized by the existence of a single UX team with a company-wide focus, serving as a supporting unit for all product divisions. The organizational structure of this UX team can be further divided into smaller functional teams focusing on specific topics and areas within the field of UX (Kaplan and Pernice 2019). UX contributors are flexibly assigned to projects for the individual products, meaning that a single contributor can work for various products over the course of time or at the same time. The assignment of the UX contributors is determined by their expertise in a specific field of UX rather than profound product knowledge (Kaplan and Pernice 2019).

Software development can benefit from a centralized UX structure in multiple ways: From product perspective, the UX consistency of all products can be ensured, as all UX effort is coming from a team reporting in a single organizational structure. Based on the comprehensive skill set with regards to all aspects of UX provided by the different UX contributors, the centralized team can always provide the most suitable contribution to the project. While this is a benefit of the centralized approach, it is also a challenge that it faces. UX contributors are lacking in-depth product knowledge and additional effort and time is necessary to provide a qualitative assessment of the user requirements for the product (Kaplan and Pernice 2019). Bringing UX to the table of product decisions can be hard in a centralized system, as the UX team is seen as a separate unit. Product divisions need to proactively request the consultation by the UX team which presupposes that the need for UX consultation and its value is recognized by the product development team first (Rosenbaum et al. 2000, 340; Kaplan and Pernice 2019). Overall, Rohn (2007) rates a centralized UX organization to be most effective for most companies (Rohn 2007, 4). Oppositional, Rosenberg (2019) assesses centralized UX to be a model of the past, that was seen as the optimal mature structure “for the first two decades of corporate UX practice” (Rosenberg 2019, 30).

Distributed Structure

Distributed structures embody a contradictory approach to the organizational integration of UX. UX contributors are distributed across the company and the product divisions. In a distributed model, the UX work force is located much closer to the development work force of the products (Rosenberg 2019, 30). Often UX contributors are structurally directly integrated into the development teams of specific products. As a result, individual UX contributors collaborate consistently with the same developers and product managers within their respective teams. For most product development teams, there are only a few UX contributors working on the same product – sometimes even only a single contributor. Consequently, the UX contributors within a distributed model need to be more versatile in terms of their UX skills. However, they develop stronger product specific domain knowledge over time (Kaplan and Pernice 2019).

The main advantage of distributed UX contributors is that they work directly at the product development level. Thus, it is easier for them to influence product decisions by having a fixed place in regular team processes. Additionally, the value of UX can be better demonstrated by product specific improvements, sparking long-lasting trust by the direct product management. With these benefits, the distributed model seams suitable for implementation in an agile setting. However, the strict timing of agile methods can limit the possible effort spent on UX topics. Disadvantages of a distributed model are also evident looking at alignment issues. Ensuring a consistent UX over multiple products and preventing redundant efforts with separated UX contributors demands sophisticated communication (Kaplan and Pernice 2019). Szóstek (2012) suggests to “provide them with a forum, place and time to meet and discuss their projects as a way to support and learn from each other” (Szóstek 2012, 13).

	Centralized	Distributed	Matrix	Federated
Number of UX contributors per team	++	○	+	+
Variety of UX capabilities	++	○	+	+
UX consistency	++	○	+	++
Visibility of UX value for product team	○	+	+	+
Product knowledge of UX contributors	○	++	++	++
UX presence in product team	○	++	++	++
UX influence on product decisions	○	++	++	++

Table 1: Effects of organizational structures for UX contributors (++ = high, + = medium, ○ = low)

Matrix Structure and Federated Structure

Matrix structures and federated structures are intended to combine the benefits of both the centralized and the distributed system, while compensating their disadvantages. Within a matrix structure, UX contributors are organizationally connected to two different managers, a product manager and a UX manager. To ensure a consistent organizational structure, one reporting line to one manager (either UX or product) is formally prioritized, while the other is rather informal (Kaplan and Pernice 2019). The main advantage of UX contributors in a matrix setting is their direct product involvement, combined with the strengths of an overarching central UX organization (Rohn 2007, 4).

Rosenberg (2019) introduces another term for an organizational structure that claims to combine the benefits of centralized and distributed UX – the federated structure. In contrast to the matrix structure, the federated model envisions multiple dedicated UX teams embedded within the product divisions, not only single contributors. These product specific UX teams are backed by a centralized UX team, responsible for company-wide visibility of UX, consistency standards, and specifically skilled UX support (Rosenberg 2019, 31).

Overall, matrix and federated settings improve the presence of UX within the product development teams, leading to higher value put on the work of UX contributors as well as better product knowledge of the UX contributors. Hence, the influence of UX input on product decisions is higher. Nonetheless, both matrix and federated models have their disadvantages. Within the matrix setting, UX contributors “may feel pulled between their two different managers” (Kaplan and Pernice 2019). Additionally, the added organizational complexity through the matrix structure can slow down decision making and demands additional alignment. Within a federated setting it can be challenging to claim the authority of the central UX team in providing standards as the product level UX teams might question their decisions based on product knowledge. UX consistency is hence more difficult to ensure (Rosenberg 2019, 34).

Applying the Fully Agile Unit Model to UX

Although there is no clear positioning of UX within the fully agile unit, based on the research on organizational structures for UX contributors, the concepts can be mapped to each other: Centralized UX teams use the concept of shared teams, supporting multiple product development teams (squads) or even multiple product areas (tribes). Distributed UX contributors on the other hand are directly incorporated into the product development teams and can hence be seen as either part of the core or extended team, depending on the UX focus of the organization. Providing them a forum (Szóstek 2012, 13) can be compared with the guild concept of the fully agile unit. Furthermore, organizing UX contributors as chapters is comparable to the matrix UX structure, having UX contributors incorporated within the product development teams but still collectively organized by a semi-central UX team. Federated UX teams, as described by Rosenberg (2019) can be implemented as separate squads within the respective product area (tribe).

Autonomy

To provide faster decisions, agile software development features the concept of self-managing individuals and self-organizing teams (Beck et al. 2001b, eleventh principle). A key feature of self-organizing teams is autonomy. Moe et al. (2008) introduce three levels of autonomy: external, internal, and individual autonomy.

External autonomy is determined by the influence of team-external parties, such as management and other individuals. The higher the influence of these, the lower the external autonomy of a team. Internal autonomy is high when the decision power within the team is shared between all team members. Decisions are not taken by e.g., a team lead, but by either the entire team or on-demand authorized subgroups. Individual autonomy refers to high individual control over the process of working on assigned tasks – characterized by few rules and constraints. Individual autonomy, although providing freedom to the individual team members, can lead to challenges with regards to internal autonomy. With self-structured work of the individuals, the frequency of interaction with other team members decreases, making it more difficult to achieve cohesive team decisions (Moe, Dingsøyr, and Dybå 2008, 78).

Moe et al. (2008) conclude that traditional software development teams are characterized by high individual autonomy but low internal and external autonomy. Agile teams on the other hand demand for a balanced amount of all three levels of autonomy. To achieve this, first, external management influence must be limited to increase the authority of the team in project decisions. Thus, team members can better identify with the project and hence foster interaction and shared decision-making. Second, there is a need for redundancy in the roles of team members to ensure that contributors can assist or substitute each other. Consequently, Moe et al. (2008) recommend that the team members acquire multiple skills to achieve functional redundancy. Although redundancy is typically considered counterproductive and uneconomical, within self-organizing teams it is crucial to achieve flexibility and support rapid decision-making (Moe, Dingsøyr, and Dybå 2008, 82–83).

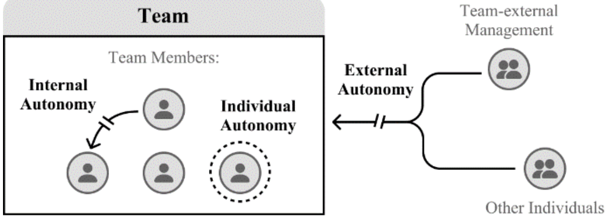

Figure 1: External, internal, and individual autonomy

Impact of Organizational Structures on Software Quality and UX

Structural organization influences the quality of software via multiple factors. Damaševičius (2010) mentions that there is a direct relation between the quality of software code produced and the team structures that produced the code. Especially when it comes to interfacing software components, they can only be of high quality if the communication between the involved contributors was working well (Damaševičius 2010, 12). This is referring to Conway's law, which in detail states that "[…] organizations which design systems […] are constrained to produce designs which are copies of the communication structures of these organizations." (Conway 1968, 31) Consequently, the determined organizational structure crucially influences the expected outcome and should hence be carefully planned and observed.

In literature, there is no fundamental evidence that organizational structure directly influences the UX perceived by the users. Nonetheless, indirect relations can be identified. Software quality directly determines the utility and usability of a software as low quality of software results in usability issues for the user. Hence, good software quality resulting amongst others from optimized organizational structures influences the perception of the software by the users and therefore determines the UX of the software. Furthermore, assuming UX contributors as part of the product development teams, Conway's law can also be applied to them. It is hence crucial that UX contributors closely collaborate both with the development team as well as with potential other UX contributors working on connected features and software components.

PROCESS DESIGN OF AGILE SOFTWARE DEVELOPMENT AND UX

Challenges

Agile software development methodologies have gained popularity due to their iterative and customer-centric approach. The user-centric and iterative approach of UX implies that agile software development and UX fit well together. However, integrating UX practices into agile projects presents several challenges. One of the main concerns highlighted by Larusdottir et al. (2012) is the lack of the big picture of UX in agile projects. Because agile software development focuses on individual features rather than holistic UX, both the UX work and its outcome can be fragmented in an agile environment (Larusdottir, Cajander, and Gulliksen 2012, 2).

Another challenge identified by Chamberlain et al. (2006) is the power struggle between UX contributors and developers (Chamberlain, Sharp, and Maiden 2006, 152). UX may be perceived as optional or peripheral to development work, leading to a lack of collaboration between UX contributors and developers (Jurca, Hellmann, and Maurer 2014, 28). Silva da Silva et al. (2012) find that UX contributors are often not perceived as full members of the product development team and work on multiple projects simultaneously. This approach to UX hinders close collaboration between UX contributors and developers (Silva da Silva et al. 2012, 749–50).

Kuusinen and Väänänen-Vainio-Mattila (2012) point out that UX work is often not included in development processes, and decisions regarding its inclusion are not made by UX specialists. As a result, UX issues are considered too late in the process, leading to inefficiencies in the use of UX resources (Kuusinen and Väänänen-Vainio-Mattila 2012, 145–47). In addition, Chamberlain et al. (2006) point out differences in time frames, communication issues, and a reluctance to understand each other's needs within the collaboration between UX and development (Chamberlain, Sharp, and Maiden 2006, 152).

To overcome the challenges mentioned above, the integration of agile software development and UX is dependent on processes that promote collaboration, coordination, and communication. The goal is to integrate the benefits of agile software development and UX without limiting either in its practices or methods.

Practices

Big Picture

Consistency throughout a product is essential for a positively perceived UX. To achieve consistency, UX contributors require a vision of the entire product in advance of creating the product design (Kuusinen 2014, 266). Agile software development, especially Scrum, on the other hand is feature-oriented. Hence, developers focus primarily on the short-term scope of the current iteration rather than the big picture of the product (Larusdottir, Cajander, and Gulliksen 2012, 6). Larusdottir et al. (2012) mention that it is more convenient for developers to work on a small fraction of the product at a time, while they are not responsible for keeping track of the big picture of the product – for which the product owner is responsible (Larusdottir, Cajander, and Gulliksen 2012, 4). However, to achieve a positively perceived UX, a collectively known big picture is crucial. Consequently, it is suggested that each role within the product development team should be responsible for keeping the product vision in mind. Kuusinen and Väänänen-Vainio-Mattila (2012) summarize this practice as "all together from early on" (Kuusinen and Väänänen-Vainio-Mattila 2012, 145).

Big or Little Design Upfront

A common perception of UX practices is that they involve "big design upfront" (Larusdottir, Cajander, and Gulliksen 2012, 6), a demand for fixed requirements and a strict handoff to development at the end of the design work, making it difficult to include in regular iterations. In contrast, Beyer (2010) argues that UX processes are in fact iterative as well (Beyer 2010, 11). Thus, the literature refers to "little design upfront" as an appropriate approach for UX within agile software development – acknowledging that certain design tasks need to be done upfront while affirming that the main UX activities can be done iteratively in parallel with the development iterations (Silva da Silva et al. 2012, 744).

Parallel and Iterative Tracks

Although UX activities can and should be performed iteratively throughout the development sprints, it is essential to allow sufficient time for UX contributors to focus on the design prior to the implementation of the respective artifact (Jurca, Hellmann, and Maurer 2014, 29). Consequently, the literature suggests shifting the main UX activities to a parallel track, one sprint ahead of the implementation sprint of the respective feature. With this method, UX contributors work on the upcoming feature one sprint ahead, while overseeing and supporting the implementation during the current sprint (Sy 2007, 118).

With this "one sprint ahead" practice in place, UX and development work is done in parallel synchronized tracks (Kieffer, Ghouti, and Macq 2017, 578). While the members of both tracks work independently on their current sprint items, regular collaboration is required to support the current implementation. Several research papers recommend the practice of parallel tracks (Larusdottir, Cajander, and Gulliksen 2012; Sy 2007; Miller 2005; Salah, Petrie, and Paige 2009). It serves to combine the benefits of regular interaction between UX and development, while allowing them the necessary autonomy to organize and pursue their activities individually.

To fulfill the purpose of having UX and development working closely together, it is essential that these parallel tracks are both synchronized and interwoven, as mentioned in the third principle of user-centered agile software development by Brhel et al. (2015) (Brhel et al. 2015, 18–19). The separation into different tracks should not lead to less interaction and collaboration between UX and development.

To support the previously introduced parallel tracks, it is necessary that UX contributors and developers work on the same sets of features. While it is common in agile software development to break large parts of the big picture into smaller pieces to work on, it is less common for UX activities. Therefore, to integrate UX into agile software development, it is expected that UX contributors will work on smaller feature sets rather than designing the entire product end to end. These small feature sets, tied to the current scope of the sprint, incrementally add up to a growing prototype of the product (Kieffer, Ghouti, and Macq 2017, 578).

In addition, the iterative nature of both agile software development and user-centered design should be encouraged, as described in the second principle of Brhel et al. (2015) (Brhel et al. 2015, 17–18). Although UX contributors create initial prototypes of the features one sprint ahead of development, these prototypes should not remain unchanged. The design should be iteratively revised both within the preceding UX sprint and within the implementation sprint. Iterative changes should be based on the regular interaction of the UX contributors with other UX contributors, developers, and other involved stakeholders such as management, customers, and users (Brhel et al. 2015, 17).

Handoff and Collaboration

Traditional software development methods rely heavily on formal handoff processes between the various contributors and development stages involved. Agile software development, on the other hand, explicitly refrains from incorporating fixed handoff processes to promote the iterative and lightweight collaboration between the different contributors and teams involved. Larusdottir et al. (2012) find that most of the collaboration between UX contributors and developers takes place through informal practices – further highlighting the need for close personal connections between the collaborating roles (Larusdottir, Cajander, and Gulliksen 2012, 6). Additionally, close connections foster early and frequent communication. Jones and Thoma (2019) advocate for UX contributors and developers to build these relationships to enable faster problem identification, problem solving, ideation, and decision making. Moreover, direct connections between UX contributors, product owners, and developers improve the understanding of each other's role, fostering mutual understanding and the sharing of skills and knowledge (Jones and Thoma 2019, 35–36).

The different roles of a product development team need effective artifacts to discuss and communicate the current state and upcoming plans (Brhel et al. 2015, 22). These artifacts should be adaptive, as opposed to fixed documentation, in line with the second value of the Agile Manifesto ("Working software over comprehensive documentation" (Beck et al. 2001a)). Bruun et al. (2018) argue that wireframes, mockups, and prototypes created by UX contributors can serve this purpose as effective and up-to-date artifacts for communication both within the team and with external stakeholders, such as customers and users (Bruun et al. 2018, 7). In addition to communicating and informing, these artifacts can also be used to actively involve developers and product management in the design of the solution, enabling iterative co-creation rather than strictly isolated delivery steps (Jones and Thoma 2019, 27).

Although agile software development focuses on the rapid creation of functional prototypes, the previously mentioned artifacts have gained popularity within agile methodology and can therefore facilitate the integration with other UX-based processes and tools (Brhel et al. 2015, 22).

CASE STUDY

Goal

To evaluate and demonstrate the relevance of the integration of UX contributors in agile software development, we conducted a case study, adding to both the relevance cycle and the design cycle of our Design Science Research approach. This case study examines how a large international software company integrates product development teams and UX contributors for a specific product area. The company has multiple product areas, and it embraces agile innovation for its products.

In terms of organizational structure, the study describes the structure of the case organization and analyzes how the UX department fits into the company's overall structure and product development teams. It compares these findings to established organizational design principles and generic agile unit and UX team structures. The study also explores organizational goals and team members' experiences related to the structure.

On the process side, the study evaluates the application of defined practices within the organization and how they are supported by managerial activities. It focuses on the collaboration between UX contributors, product owners, and developers, while also assessing the priority of UX within the organization and its impact on UX contributor integration.

Method

Referring to the Design Science Research method, this case study serves as a practical context to further develop, evaluate, and refine the findings and concepts from the literature presented earlier.

The case study is primarily qualitative, gathering insights from experts and practitioners in user-centered agile software development through five interviews. These interviews spanned various roles, including UX management (two UX managers), UX practice (a UX lead), product management (a product owner), and development (a developer). Given the global distribution of interviewees, the interviews were conducted online and were semi-structured, with tailored key questions.

For UX managers in Germany and the USA, the interviews focused on organizational design and expectations for UX, development, and product management collaboration. The interview with a USA-based UX lead centered on practical application of processes and their impact on role coordination. A Canadian product owner's interview explored product management's UX integration expectations. The interview with a German front-end developer delved into working with UX contributors and how organizational structures affect role collaboration. All interviewees were questioned about their learning experiences from working with other roles.

Additionally, a workshop and group discussion involved eight UX contributors from various locations (Germany, Canada, Poland, China). It dealt with preferred organizational structures and experiences with integrating UX and agile development practices. Furthermore, the study examined relevant documents outlining processes and practices and was complemented by observations of departmental activities.

Findings

Structural Organization

Product owners as well as development teams of the specific products within the product area are organizationally cumulated under the individual heads of product. Each head of product directly reports to the chief product officer of the product area. Structurally, UX is not part of the product development teams. All UX contributors for the different products are combined in a separate product area specific UX department under the head of UX who directly reports to the chief product officer of the product area.

Within the UX department, the UX contributors are grouped by their different locations (Northern America, Europe, Asia), each having a respective UX manager. In addition, individual contributors act as UX leads for the different products. These directly report to the head of UX. UX contributors are structurally reporting to their location managers but work in teams for the different products, led by the respective UX leads. Product area specific UX departments are common across the entire company. Together with a central UX department, they form a federated UX structure. The central UX department acts as a supporting unit.

The structural integration of the UX contributors in the considered case does not follow a specific structure of the fully agile model but applies a combination of its characteristics. Within the UX department, UX contributors form distinct groups, similar to squads, led by their location-specific UX managers. However, their main activities are performed within chapters that focus on individual products and are led by the UX leads. Additional guilds for specialized topics can be found within the UX department. In the context of the separated product development teams, the UX department can be classified as a supporting shared team.

In terms of autonomy, the interviewed UX managers emphasize that a certain amount of external autonomy is beneficial. Considering the different locations of the UX managers and teams, external and internal autonomy allows them to manage their teams in a way that best fits the team's culture and way of working. In addition, internal autonomy allows product-specific UX teams to collaborate on solutions without being dependent on the decision of individual managers. As a result, the UX lead and UX managers see their responsibility in enabling their teams to make decisions together, rather than driving the decisions themselves. Increased external autonomy, on the other hand, invokes the feeling of siloed departments, as criticized by the UX contributors. To achieve organizational goals and product goals, the focus should be on the collaboration and connectivity of the teams and contributors involved. According to one of the

UX managers, the balance lies in having autonomy for the team while also incorporating a connected approach that brings together the right expertise and perspectives.

According to the UX lead, having a dedicated UX department for the product area improves the influence of UX and helps align UX efforts with business goals. Because UX workloads vary across products, the flexible matrix structure of the UX department allows UX managers to quickly adjust the assignments of UX contributors. Being part of a larger UX department enables the UX contributors to collaborate on solutions. Additionally, having the smaller UX teams for each product promotes quick decision-making and increases organizational agility. While these factors highlight the benefits of having a separate UX department within the product area, the interviewees also express a desire for closer proximity between UX contributors and development teams. The product owner believes that integrating UX effectively into the development lifecycle requires the UX contributors to be as close to the development teams as possible. In the current structure, UX contributors report that they do not feel part of the product development teams. The consulted developer mentions that having UX contributors on the same team would significantly improve feedback, allowing for easier clarification and better agreement on realistic requirements.

Processes and Practices

Big Picture: The interviewed UX lead mentions that focusing on the big picture in advance helps to streamline the development process. In practice, the UX contributors experience contradicting realities. They report that some product development teams involve UX early in the process while others start to consult UX at a very late stage. While the former leads to a better understanding of the product and helps to identify gaps of the product vision early, the latter results in a lack of the big picture and hence in conflicting UX. The UX contributors further argue that missing a collective product vision leads to ever-changing requirements due to misalignment within the product development teams. As a result, in some cases, UX contributors report to work on features without having a clear definition of the use case.

According to one of the interviewed UX managers, UX involvement in the early envisioning phases of new products or projects is often limited and needs to be improved. This impression coincides with the experience of the UX lead who mentions that in early product envisioning phases, UX is excluded from most strategy planning. This disconnect consequently leads to problems when product owners ask for a comprehensive look at the product appearance without involving UX from the beginning, as the UX lead reports. The other interviewed UX manager also confirms that involvement in the envisioning phase should be cross-functional. Previous projects that were either entirely driven by product management or entirely driven by UX in their envisioning phases lacked either user-centricity or technical feasibility. The product owner emphasizes the benefits of involving UX early in the process, not only for improving the usability of the product, but also for refining requirements. However, the product owner further mentions that "[…] it is currently only a matter of personal preference if [product owners] want to work with UX or not."

One Sprint Ahead: During the interview, the UX lead recalls former processes where UX work was integrated directly into the development sprints, working on the same features as development during the same sprint. As this approach caused stress and dissatisfaction among the UX contributors, the developers as well as the product owners, the lead time of UX activities ahead of development was increased. Both the product owner and the UX lead concurringly report that currently UX activities take place not only one but multiple sprints ahead of development – usually four to six weeks in advance of the implementation, allowing for sufficient refinement and iterations. For the product owner, this added time allows for better capacity planning and it ensures that the developers are equipped with sophisticated UX artifacts when the implementation begins.

Parallel and Iterative Tracks: According to the interviews conducted, the UX and development work does not have a specific synchronization with respect to the development sprints. Nevertheless, a product area specific product design and development process features several touch points to support the synchronicity of the UX and development tracks. In practice, UX contributors experience alignment issues. For instance, features are released without a final UX sign-off performed. One interviewee states that sometimes small improvements and compromises get lost, cumulatively impacting the UX of the product. The product owner and UX contributors identify limited personnel and time resources as a reason leading to these process deviations. Overall, iterative handoffs between UX and development are common. Even during the implementation sprint, adjustments are made to the previously created designs. While regular change is characteristic for agile software development, both UX contributors as well as developers sometimes report this as unexpected rework. The product owner mentions: "Developers prefer the path of least resistance. They don't want to refactor what they have done before. That's normal. […] So, you have to be prepared to explain the reasons for the change." As a result of this, UX improvements that do not directly impact the functionality may be rejected because of the effort required to implement them. The UX lead concurringly mentions that UX improvements are difficult to achieve unless they were included in the initial design phase. This leads to a perceived lack of UX investment in certain products.

Collaboration: Two main styles of collaboration are evident during the case study: formal and informal collaboration. The formal collaboration mainly entails fixed regular meetings. Additionally, a significant part of the collaboration is supported by informal means of communication. These include ad-hoc calls between UX contributors and developers as well as conversations via chat messages for quick discussions. This proofs to be particularly valuable when facing roadblocks during development, as mentioned by the developer. Informal collaboration also resonates well with the product owner, who mentions quick refinement of requirements in close collaboration with UX contributors. All consulted roles agree that personal connections and relationships are key to successful collaboration. Especially the UX lead and the UX managers highlight the importance of a deep connection and close relationships between UX contributors, product owners, and developers. Without these strong relationships UX is often rather seen as a blocker than a benefit, as the UX lead has experienced. Therefore, the UX lead encourages UX contributors to build up strong partnerships with their respective developers.

Besides building strong relationships, another enabler for effective collaboration mentioned by the interviewees is gaining general knowledge about the respective other roles that they are working with. The UX contributors report that collaboration especially works well when the responsible product owner is well-versed around UX and consequently values UX practices. The interviewed product owner expresses the desire to have a role within the development teams to "champion" UX topics to convey the importance of UX among the developers. The UX lead also emphasizes the importance of developers understanding the value of good UX and the need to work hand in hand to achieve it. Since developers are initially not trained regarding the practices of UX contributors and vice versa, it takes time for the roles involved to adapt and to effectively collaborate, as stated by the interviewed developer.

UX Priority: According to both UX managers, UX is considered very important in the organization, but there are challenges in fully integrating it into the product development. The main challenge identified is the partly low prioritization of UX within the product development processes. The priority of UX related work is mainly influenced by the respective product owners. The interviewed product owner as well as the UX contributors and the UX lead note that the influence of UX within the different product development teams varies significantly due to the different preferences of the individual product owners. The UX lead summarizes: "UX is often considered important in theory but then tends to be deprioritized in practice."

The interviewed product owner considers UX a top priority but acknowledges that prioritizing UX too high can also result in challenges of deprioritizing non-UX related tasks. The UX contributors furthermore mention that UX related processes such as the product area internal product design and development process are mainly enforced within the UX team but not fully adopted by the product development teams. Hence, UX contributors wish for a more stable representation within the processes of the product development teams.

FRAMEWORK OF INFLUENTIAL FACTORS AND IMPLICATIONS

To design a framework, we aimed to combine the insights from our literature research (knowledge base) and case study (environment). We therefore clustered the insights to visualize connections and contradictions between literature and practice and to identify mutual relations between the researched aspects. Based on these clusters we consolidated the insights into actionable factors, iteratively evaluated by the experience from the case study (design cycle of the Design Science Research methodology).

As a result of the consolidation, we identify the following as the main factors of the organizational integration of UX contributors in agile software development:

(1) Organizational structure of agile product development teams and UX contributors.
(2) Degree of autonomy of the teams involved.
(3) Priority given to UX in agile software development processes.
(4) Agility of UX processes and practices.
(5) Mutual integration of agile software development and UX processes.
(6) Mutual knowledge about the roles involved.
(7) Influence of collaboration and relationships between the roles involved.

It can be said that the identified factors are either structure-related or process-related. Furthermore, they are either characteristics of UX, of agile software development, or of the interface between the two. The identified factors contribute to the success of the integration by improving the effectivity and efficiency of software development.

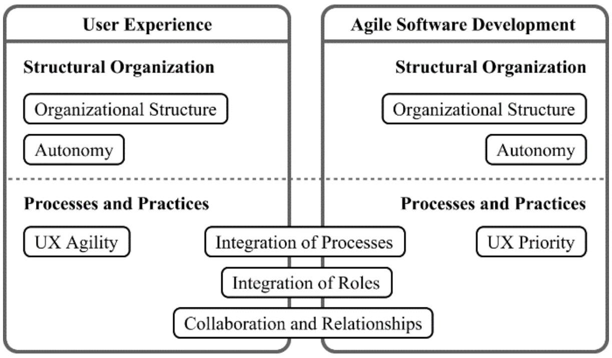

Figure 2: Influential factors for the organizational integration of UX contributors in agile software development

Structural Organization

(1) Organizational Structure – An organizational design that allows for the addition of collaborative structures

serves as a foundation for the effective integration of UX contributors in agile software development:
The fully agile unit, consisting of tribes and squads, serves as a systematic model to represent the generic structure of agile software development teams (Gonçalves and Lopes 2014). With the concepts of chapters, guilds and shared teams, the fully agile unit provides multiple ways to integrate the core product development teams with additional structures.

The structural integration of UX contributors requires a balance between the comprehensiveness of shared UX teams and the sense of direct involvement of UX contributors in product development teams:
UX contributors are not specifically represented within the fully agile unit model. However, a set of typical organizational structures with regards to UX is introduced within this paper. These include centralized, distributed, matrix, and federated approaches (Kaplan and Pernice 2019; Rohn 2007; Szóstek 2012; Rosenberg 2019). The evaluation of these structures shows advantages and disadvantages of each of them when integrated into agile software development. The centralized approach facilitates consistency and broad UX skills but limits in-depth product knowledge and agility. While distributed UX contributors across different product development teams can compensate for these disadvantages by being deeply involved in the product, their ability for consistency and breadth of UX activities is limited.

Matrix structures and federated approaches aim to combine the advantages of both by applying centralized as well as distributed aspects to the organizational structure. Especially in terms of agility, they benefit from having UX contributors or smaller UX teams directly integrated in product development teams or product areas. Their ability to quickly adapt to changing product requirements is enhanced by deep product knowledge. At the same time, the individual UX contributors benefit from the support of a larger group of UX contributors around them. However, using a matrix or federated structure increases the complexity of the organizational structure when viewed in the context of the product development team structure. The case study shows that the organizational design of product development teams and UX teams can differ substantially. Furthermore, the case study shows that a federated structure consisting of product area specific UX teams can limit the actual integration of UX contributors into the product development teams by reducing the sense of belonging.

Finally, both the theoretical research and the conducted case study show that the different organizational structures regarding UX all have advantages and disadvantages. Based on the contrasting characteristics of the centralized and distributed approaches, it is advisable to adopt a structure that combines both approaches. However, it is then crucial to maintain a balance between deep integration and supportive central backing.

(2) Autonomy – Balancing autonomy and connectedness of the involved teams enables effective and efficient collaboration:
Moe et al. (2008) introduce autonomy as a characteristic of self-organizing teams, common to agile software development. They recommend a balanced amount of external, internal, and individual autonomy for agile software development teams (Moe, Dingsøyr, and Dybå 2008, 82–83). However, based on the findings of the case study, autonomy should always be balanced with connectedness to foster collaboration and prevent the emergence of silos. Regarding the integration of UX and agile software development, the case study shows that external autonomy is beneficial in terms of enabling the different working styles of the teams involved. However, as the integration of UX contributors with the product development teams is critical, autonomy should not limit the connectivity of the roles involved. Since external autonomy can amplify the feeling of disconnection, it should be balanced with shared processes. Internal and individual autonomy, on the other hand, support collaborative decision-making within the teams involved and should therefore be encouraged.

Processes and Practices

(3) UX Priority – The prioritization of UX by product management enables the effective involvement of UX contributors in the software development:
Since none of the major agile software development methodologies explicitly incorporate UX contributors, pure agile software development processes do not reserve time and resources for involving UX practices (Kane 2003, 1). To integrate UX, it is therefore necessary to prioritize UX tasks within the entirety of tasks performed during the development process. The case study shows that this UX priority is mainly influenced by the product owners, who determine the criticality of the tasks to be performed. UX should hereby not fall behind other development tasks. While organizations with a high level of UX maturity naturally set a high priority to UX, organizations with a lower level of UX maturity tend to deprioritize UX due to time and resource constraints. Without a high UX priority, UX contributors are limited in their ability to influence the product. In addition to product management, UX management can also influence the prioritization of UX by raising awareness of the importance of UX.

(4) UX Agility – Incorporating agile considerations such as iterative and incremental design into the UX processes enables UX contributors to efficiently contribute to an agile software development process:
While improvements in UX prioritization help bring agile software development closer to UX considerations, UX contributors should also incorporate agile considerations into the way they work. As Kieffer et al. (2017) point out, UX contributors are used to creating a design for the entire product. Working on tightly constrained features instead, as is common in agile software develop-

ment, requires process changes in the way UX contributors work (Kieffer, Ghouti, and Macq 2017, 578). While the big picture of the product vision should still be top of mind, iterations should only focus on the current solution to a specific identified problem. Furthermore, for agile processes it is crucial to acknowledge that requirements can change frequently and even at a late point in time during development, as mentioned in the second principle of the Agile Manifesto (Beck et al. 2001b). The case study highlights that this frequency of change can be unfamiliar to UX contributors, who may feel overwhelmed by changing requirements. However, the iterative nature of UX processes supports the increasing agility of UX activities (Beyer 2010, 11).

(5) Integration of Processes – Combined practices enable UX, product management and development to efficiently co-create the product while maintaining process agility:
The theoretical research of this paper shows that specific processes and practices should be considered for the integration of UX in agile software development.

Emphasizing a common big picture of the product to be developed improves the outcome of the collaboration between UX contributors, product management, and development. While agile methods focus primarily on short-term feature development (Larusdottir, Cajander, and Gulliksen 2012, 6), an overall product vision is necessary to achieve a consistent and thus positively perceived UX of the product (Kuusinen 2014, 266). This not only enables the UX contributors to create consistent designs, but also supports the product management in defining coherent features and precise requirements as the case study shows. The vision should be developed jointly by all involved roles, keeping them engaged from early on. This also encourages all roles to feel equally involved and to identify with the product – increasing their engagement and consequently the quality of the outcome (Chamberlain, Sharp, and Maiden 2006, 152).

To improve the direct collaboration between UX contributors and development, their individual processes must be interwoven (Brhel et al. 2015, 17–19). Working in parallel tracks on the same feature sets helps achieve this (Larusdottir, Cajander, and Gulliksen 2012; Sy 2007; Miller 2005; Salah, Petrie, and Paige 2009). To ensure effective transfer of UX considerations into implementation, the research recommends giving UX contributors a lead time to prepare designs, ideally one sprint ahead of the implementation (Sy 2007, 118). In practice, as the case study shows, these lead times for UX can also be extended to provide additional time for thorough research and ultimately improve the quality of the designed solution. However, excessively extending the time difference between UX work and development can hinder the synchronicity between the two tracks and consequently reduce the agility of the process, as retrieved from the case study.

During their parallel work, UX contributors and developers focus on the same feature sets derived from the product vision. Agile principles are applied through iterative and incremental approaches (Kieffer, Ghouti, and Macq 2017, 578). However, the case study indicates that iterative changes can sometimes be perceived negatively by both developers and UX contributors. In particular, unsubstantiated requests for change or frequently changing requirements frustrate the UX contributors. Requests for design changes after the implementation furthermore frustrate developers. To mitigate frustration, it is important for UX contributors and developers to be open to change and for product owners to thoroughly evaluate and explain changing requirements.

(6) Integration of Roles – Improving the mutual knowledge about collaborative roles within product development teams improves the effectiveness of the collaboration:
Both agile software development and UX employ specifically defined roles, introduced in this paper. When it comes to integrating the roles of UX contributors with the agile software development environment, these predefined roles must work jointly. According to Jones and Thoma (2019), a thorough understanding of each other's role significantly improves the ability to collaborate (Jones and Thoma 2019, 35–36). This is further supported by the findings of the case study, which emphasize that UX contributors, product owners, and developers must first understand each other's roles in order to develop effective ways to support each other and utilize the benefits of each other's work.

(7) Collaboration and Relationships – Informal organization in the form of strong individual relationships between the collaborating roles increases the efficiency and effectiveness of the collaboration and compensates for possible compromises in the formal organization:
Theory emphasizes that traditional software development relies on formal handoff processes, while agile development favors iterative and lightweight collaboration between contributors and teams. Larusdottir et al. (2012) find that in agile environments, most collaboration between UX contributors and developers occurs through informal practices, emphasizing the need for close personal connections among team members (Larusdottir, Cajander, and Gulliksen 2012, 6). Furthermore, the fourth principle of the Agile Manifesto states that "business people and developers must work together daily throughout the project" (Beck et al. 2001b). Based on the insights from this paper, this principle should also be applied to additional roles such as UX contributors, working with both business and development. Establishing these connections leads to early and frequent communication and enables faster problem identification, problem solving, idea generation, and decision making between UX contributors and developers.

In practice, the case study confirms two main styles of collaboration: formal and informal. Formal collaboration involves regular meetings where product owners, developers, and UX contributors come together to exchange on the current state of their work. The case study highlights that informal communication plays a significant role in supporting collaboration. Ad-hoc meetings and chat messages allow for quick efficient discussions, particularly beneficial when fast decisions are required during iterations. Furthermore, close connections allow for rapid refinements of requirements through informal collaboration between UX contributors and product owners.

It can be concluded that personal connections and relationships are essential for successful collaboration across different roles, especially when located in structurally separated teams. Strong relationships between UX contributors, product owners, and developers are crucial for overcoming the perception of UX as an obstacle rather than a valuable asset.

Overall, the collaboration should not be overly dictated by strict processes. Strong interpersonal relationships should take precedence over strict adherence to rigid organizational structures. Especially in a federated organizational structure, effective communication plays a key role in successful collaboration across multiple teams involved. Prioritizing close collaboration is further supported by the first value of the Agile Manifesto, which values “individuals and interactions over processes and tools” (Beck et al. 2001a). More precisely, the sixth principle of the Agile Manifesto states that “the most efficient and effective method of conveying information to and within a development team is face-to-face conversation” (Beck et al. 2001b). The research shows that this principle also applies to the collaboration between UX contributors, product owners, and developers.

Levels of Organizational Influence

The identified factors can be influenced either on an individual, team-internal, or team-external level. While the organizational structure of both the agile product development teams as well as the UX contributors is mainly predetermined by the structure of the wider product area (external organization), the specific integration of processes and roles can be influenced by the management of the individual teams (team organization). Furthermore, the priority given to UX, the agility of UX and the autonomy of the involved teams are heavily influenced by the directly involved team management (team organization). Lastly, collaboration and building relationships among the involved contributors from product management, development and UX directly depends on the individual involved team members and can hence be classified as an aspect of self-organization.

This classification shows that the integration is not only determined by external organization but can be substantially influenced by the direct management (UX and product management) and further by the individuals working in the teams. As the research shows, especially self-organization, focusing on building relationships and fostering direct collaboration, crucially improves the integration of UX contributors and agile product development teams.

Application as Critical Success Factors

To evaluate the usefulness of this framework with regards to the practical management of software development organizations, we draw an interrelation to the concept of critical success factors. According to Rockart (1979), critical success factors are “the limited number of areas in which results, if they are satisfactory, will ensure successful competitive performance for the organization” (Rockart 1979). Hence, these factors must be focused and continually measured by management (Rockart 1979).

Our process of deriving and evaluating the influential factors for the successful integration of UX contributors in agile software development can be compared to the process of deriving critical success factors (described by Cooper 2008, 4). The process consists of an introductory workshop, interviews, and a focusing workshop for consolidation. Based on our research, the influential factors that we outline are in fact crucial for the success of an integration. Necessary management focus is especially expressed in factors such as UX priority, UX agility and integration of processes and roles. Continuous measurement of these factors seems beneficial for the overall success. Therefore, we can infer that our identified influential factors are actionable factors that support the successful integration of UX contributors into agile software development. However, to completely fulfill the requirements of critical success factors, according to Rockart (1979), a proof of extensiveness of the identified factors needs to be researched.

CONCLUSIONS

Synthesizing our research with regards to the objectives set at the beginning, we can conclude the following:

What is state of the art for integrating UX in agile software development?

Agile methodologies are widely used in the field of software development. With the increasing awareness of the importance of UX in this field, literature research shows that organizations now face challenges in effectively enhancing agile software development with the benefits of UX considerations. Our research confirms that the principles of agile software development and UX are indeed compatible. However, to effectively incorporate UX considerations into agile software development methodologies, organizational aspects of integrating UX contributors and agile product development teams need to be considered. Since prominent agile methodologies do not include UX contributors in their models, new models for the integration need to be defined. To start the development of such models, this paper has investigated the factors that need to be considered for the integration.

Which organizational factors influence the integration of UX contributors in agile software development?
Organizational theory indicates two perspectives on organization: the structural perspective and the process perspective. Throughout this paper, this distinction proved to be appropriate to explore the factors of the organizational integration in a structured manner. In terms of structural factors, our research has shown that organizational design has a direct impact on the integration of UX contributors in agile software development. In particular, the organizational structures concerning the product development teams and UX contributors, the roles involved, and the intended autonomy of these were identified as the main aspects of the integration in terms of structural organization. From a process perspective, our research identified several areas of practices that need to be considered to support an effective integration. These include the product vision (jointly developing a big picture, aiming for little design upfront), synchronicity of UX and development work (parallel and iterative tracks), and process agility. Collaboration was identified as a critical factor, with formal and informal approaches emphasized in both the literature research and the case study.

Which implications for the integration of UX contributors in agile software development can be derived from these factors?
The aspects identified before were used to create a framework of influential factors for the integration of UX contributors in agile software development. Based on the findings from the literature review and the case study, the factors were enhanced with specific implications. Recognizing that the application of organizational structures always balances benefits and drawbacks, our research shows that integrated processes and practices can effectively mediate these drawbacks and compensate for compromises that must be made in the rigid organizational structure. The main areas that these processes should cover were derived from principles set in literature. The case study mainly confirmed the usage of these practices in practice. Finally, the research showed that organizational structure does not only cover factors that can be influenced by management or are externally provided, but there is a significant aspect of self-organization involved. Both, the theoretical research and the case study showed that individually established informal relationships between the roles involved significantly improves the effectiveness of the integration of UX contributors and agile product development teams.

Limitations and Future Research

To ensure the broader applicability of the framework, the factors and implications found in this paper should be validated through additional studies. The case study conducted was limited to one particular integration of UX contributors in a specific organizational setting. Thus, aspects of future research should entail to conduct similar studies in additional organizations, covering different organizational conditions. Furthermore, the amount and diversity of the consulted participants in such studies should be increased to ensure statistic generalizability. With the factors and implications validated and further enhanced through additional studies, the herein developed framework can then comprehensively support practitioners to improve the organizational integration of UX contributors in agile software development.

LITERATURE

Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, et al. 2001a. “Manifesto for Agile Software Development.” 2001. http://agilemanifesto.org/.

———. 2001b. “Principles behind the Agile Manifesto.” 2001. http://agilemanifesto.org/principles.html.

Beyer, Hugh. 2010. *User-Centered Agile Methods*. Penn State: Morgan & Claypool Publishers.

Brhel, Manuel, Hendrik Meth, Alexander Maedche, and Karl Werder. 2015. “Exploring Principles of User-Centered Agile Software Development: A Literature Review.” *Information and Software Technology* 61 (May). https://doi.org/10.1016/j.infsof.2015.01.004.

Bruun, Anders, Marta Kristin Larusdottir, Lene Nielsen, Peter Axel Nielsen, and John Stouby Persson. 2018. “The Role of UX Professionals in Agile Development: A Case Study from Industry.” In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*, 352–63. Oslo Norway: ACM. https://doi.org/10.1145/3240167.3240213.

Chamberlain, Stephanie, Helen Sharp, and Neil Maiden. 2006. “Towards a Framework for Integrating Agile Development and User-Centred Design.” In *Extreme Programming and Agile Processes in Software Engineering*, edited by Pekka Abrahamsson, Michele Marchesi, and Giancarlo Succi, 4044:143–53. Berlin, Heidelberg: Springer. https://doi.org/10.1007/11774129_15.

Chapman, Lorraine, and Scott Plewes. 2014. “A UX Maturity Model: Effective Introduction of UX into Organizations.” In *Design, User Experience, and Usability. User Experience Design Practice*, edited by Aaron Marcus, 12–22. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07638-6_2.

Cockburn, Alistair, and Jim Highsmith. 2001. “Agile Software Development, the People Factor.” *Computer* 34 (11): 131–33. https://doi.org/10.1109/2.963450.

Conway, Melvin E. 1968. “How Do Committees Invent?” *Datamation Magazine* 14 (4): 28–31.

Cooper, Vanessa. 2008. “The Critical Success Factor Method: A Review and Practical Example.” In .

Damaševičius, Robertas. 2010. “On The Human, Organizational, and Technical Aspects of Software Development and Analysis.” In *Information Systems Development: Towards a Service Provision Society*, edited by George Angelos Papadopoulos, Wita Wojtkowski, Gregory Wojtkowski, Stanislaw

Wrycza, and Jože Zupancic, 11–19. Boston, MA: Springer US. https://doi.org/10.1007/b137171_2.

Dima, Alina Mihaela, and Maria Alexandra Maassen. 2018. “From Waterfall to Agile Software: Development Models in the IT Sector, 2006 to 2018. Impacts on Company Management.” *Journal of International Studies* 11 (2): 315–26. https://doi.org/10.14254/2071-8330.2018/11-2/21.

Ferreira, Jennifer, Helen Sharp, and Hugh Robinson. 2012. “Agile Development and User Experience Design Integration as an Ongoing Achievement in Practice.” In *2012 Agile Conference*, 11–20. Dallas, TX: IEEE. https://doi.org/10.1109/Agile.2012.33.

Gerster, Daniel, Christian Dremel, Walter Brenner, and Prashant Kelker. 2020. “How Enterprises Adopt Agile Forms of Organizational Design: A Multiple-Case Study.” *ACM SIGMIS Database: The DATABASE for Advances in Information Systems* 51 (January): 84–103. https://doi.org/10.1145/3380799.3380807.

Gerster, Daniel, Christian Dremel, and Kelker Prashant. 2018. “Scaling Agility: How Enterprises Adopt Agile Forms of Organizational Design.” In . San Francisco, CA.

Gonçalves, Ester, and Eurico Lopes. 2014. “Implementing Scrum as an IT Project Management Agile Methodology in a Large Scale Institution.” In , 461. Academic Conferences International Limited.

Gray, Colin M., Austin L. Toombs, and Shad Gross. 2015. “Flow of Competence in UX Design Practice.” In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 3285–94. CHI ’15. New York, NY: Association for Computing Machinery. https://doi.org/10.1145/2702123.2702579.

Hevner, Alan, and Samir Chatterjee. 2010. “Design Science Research in Information Systems.” In *Design Research in Information Systems*, by Alan Hevner and Samir Chatterjee, 22:9–22. Integrated Series in Information Systems. Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-5653-8_2.

Hevner, Alan, Salvatore March, Jinsoo Park, and Sudha Ram. 2004. “Design Science in Information Systems Research.” *MIS Quarterly* 28 (1): 75–105.

ISO 9241-210:2019. 2019. “Ergonomics of Human-System Interaction – Part 210: Human-Centred Design for Interactive Systems.” International Organization for Standardization.

Jones, Alexander, and Volker Thoma. 2019. “Determinants for Successful Agile Collaboration between UX Designers and Software Developers in a Complex Organisation.” *International Journal of Human–Computer Interaction* 35 (20): 1914–35. https://doi.org/10.1080/10447318.2019.1587856.

Jurca, Gabriela, Theodore D. Hellmann, and Frank Maurer. 2014. “Integrating Agile and User-Centered Design: A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX.” In *2014 Agile Conference*, 24–32. Kissimmee, FL: IEEE. https://doi.org/10.1109/AGILE.2014.17.

Kane, D. 2003. “Finding a Place for Discount Usability Engineering in Agile Development: Throwing down the Gauntlet.” In *Proceedings of the Agile Development Conference, 2003. ADC 2003*, 40–46. Salt Lake City, UT: IEEE. https://doi.org/10.1109/ADC.2003.1231451.

Kaplan, Kate, and Kara Pernice. 2019. “Where Should UX Report? 3 Common Models for UX Teams and How to Choose Among Them.” Nielsen Norman Group. 2019. https://www.nngroup.com/articles/ux-team-models/.

Kashfi, Pariya, Robert Feldt, and Agneta Nilsson. 2019. “Integrating UX Principles and Practices into Software Development Organizations: A Case Study of Influencing Events.” *Journal of Systems and Software* 154 (August): 37–58. https://doi.org/10.1016/j.jss.2019.03.066.

Kieffer, Suzanne, Aissa Ghouti, and Benoit Macq. 2017. “The Agile UX Development Lifecycle: Combining Formative Usability and Agile Methods.” *Hawaii International Conference on System Sciences 2017 (HICSS-50)*, January. https://aisel.aisnet.org/hicss-50/cl/hci/7.

Kniberg, Henrik, and Anders Ivarsson. 2012. “Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds.” http://www.agileleanhouse.com/lib/lib/People/HenrikKniberg/SpotifyScaling.pdf.

Kuusinen, Kati. 2014. “Improving UX Work in Scrum Development: A Three-Year Follow-Up Study in a Company.” In *Human-Centered Software Engineering*, edited by Stefan Sauer, Cristian Bogdan, Peter Forbrig, Regina Bernhaupt, and Marco Winckler, 259–66. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-662-44811-3_17.

Kuusinen, Kati, and Kaisa Väänänen-Vainio-Mattila. 2012. “How to Make Agile UX Work More Efficient: Management and Sales Perspectives.” In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, 139–48. NordiCHI ’12. New York, NY: Association for Computing Machinery. https://doi.org/10.1145/2399016.2399037.

Larusdottir, Marta, Åsa Cajander, and Jan Gulliksen. 2012. “The Big Picture of UX Is Missing in Scrum Projects.” *CEUR Workshop Proceedings* 922 (January): 42–48.

McInerney, Paul, and Frank Maurer. 2005. “UCD in Agile Projects: Dream Team or Odd Couple?” *Interactions* 12 (6): 19–23. https://doi.org/10.1145/1096554.1096556.

Miller, L. 2005. “Case Study of Customer Input for a Successful Product.” In *Agile Development Conference (ADC’05)*, 225–34. Denver, CO: IEEE Comput. Soc. https://doi.org/10.1109/ADC.2005.16.

Moe, Nils Brede, Torgeir Dingsøyr, and Tore Dybå. 2008. “Understanding Self-Organizing Teams in

Agile Software Development.” In *19th Australian Conference on Software Engineering (Aswec 2008)*, 76–85. Perth. https://doi.org/10.1109/ASWEC.2008.4483195.

Nagappan, Nachiappan, Brendan Murphy, and Victor Basili. 2008. “The Influence of Organizational Structure on Software Quality: An Empirical Case Study.” In *Proceedings of the 13th International Conference on Software Engineering - ICSE ’08*, 521. Leipzig: ACM Press. https://doi.org/10.1145/1368088.1368160.

Rockart, John F. 1979. “Chief Executives Define Their Own Data Needs.” *Harvard Business Review* 57 (2): 81–93.

Rohn, Janice Anne. 2007. “How to Organizationally Embed UX in Your Company.” *Interactions* 14 (3): 25–28. https://doi.org/10.1145/1242421.1242440.

Rosenbaum, Stephanie, President Janice, Anne Rohn, and Director Humburg. 2000. “A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys.” *CHI Letters* 2 (1): 337–44. https://doi.org/10.1145/332040.332454.

Rosenberg, Daniel. 2019. “The Business of UX Management.” *Interactions* 26 (3): 28–35. https://doi.org/10.1145/3318131.

Salah, Dina, Helen Petrie, and Richard F Paige. 2009. “Towards a Framework for Integrating User Centred Design and Agile Software Development Processes.” In *Extreme Programming and Agile Processes in Software Engineering*, edited by Abrahamsson, P., Marchesi, M., Succi, G. Vol. 4044. Berlin, Heidelberg: Springer.

Silva da Silva, Tiago, Milene Selbach Silveira, Frank Maurer, and Theodore Hellmann. 2012. “User Experience Design and Agile Development: From Theory to Practice.” *Journal of Software Engineering and Applications* 05 (10): 743–51. https://doi.org/10.4236/jsea.2012.510087.

Sy, Desirée. 2007. “Adapting Usability Investigations for Agile User-Centered Design.” *Journal of Usability Studies* 2 (3): 112–32.

Szóstek, Agnieszka. 2012. “A Look into Some Practices behind Microsoft UX Management.” In *CHI ’12 Extended Abstracts on Human Factors in Computing Systems*. Vancouver, BC: ACM. https://dl.acm.org/doi/abs/10.1145/2212776.2212833?casa_token=SO8cRAduSLIAAAAA:g7gWihi0qVlD9lB5GHvZkgdaOOXUHtujzTF71VvBT1Iz0Qx_BanL_KUPWB-CPJ8cfk2N713qXNgk.