
Agility for UX and Development: A Case Study

John Stouby Persson

Aalborg University
Aalborg Ø, 9220, Denmark
john@cs.aau.dk

Anders Bruun

Aalborg University
Aalborg Ø, 9220, Denmark
bruun@cs.aau.dk

Peter Axel Nielsen

Aalborg University
Aalborg Ø, 9220, Denmark
pan@cs.aau.dk

Abstract

Agility characterizes practices, such as Scrum, that help a software team or organization deal with change. However, what may help a software developer to be agile may not help a user experience (UX) consultant to be agile in the same way and vice versa. Against this backdrop, we present a case study of how particular practices of integrating UX activities with agile development contribute to agility in a Danish software company. We use a theoretical framework that defines agility to analyze interviews conducted with UX professionals, developers, and managers in the company. The analysis shows how integrations of UX into agile development by upfront-design and work-in-parallel increase the agility differently for developers and UX consultants.

Author Keywords

Agility; Software development; Case study; User experience.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Verdana 7 point font. Please do not change the size of this text box.

Each submission will be assigned a unique DOI string to be included here.

Claims on integrating development and UX (Wale-Kolade et al. 2013)

1. Conduct some upfront design activities prior to project start
2. Design low-fi prototypes as the basis for developing the system
3. Perform testing in between iterations
4. Designers and developers each work in parallel
5. Usability designers should be a part of the development team
6. Usability designers should be fully integrated into the development team
7. End users or their proxies should be involved in the project life cycle

Introduction

The literature on agile software development is massive and includes much research. Despite this, there are few links to research on agility for other professions and activities than software development. Other professionals that are part of a software project, but specialize in areas different from the particular focus of developing software, may struggle to exploit agile methods. This challenge is particularly prevalent for usability and UX despite their importance to the success of many software projects (Wale-Kolade & Nielsen 2016). Researchers and practitioners may exacerbate this challenge when ascribing the agility of a software project to the use of particular methods such as Scrum or Extreme Programming (Conboy 2009). A method labelled to be agile may be practiced in a way not creating flexibility, leanness, and adaptability for at least some of participants in the software project. Thus, we may need to stop approaching agility as simply a characteristic of a method being labelled as agile or the use of elements from such methods. Instead, we could see agility as a positive effect from a practice (UX or software development) in a particular setting.

In this paper, we apply a general understanding of agility in a case study of UX activities and how they are integrated into agile development. A general definition of agile, not tied to following a particular method, is from Conboy's structured review of literature on the concept of agility across different disciplines. We use this theory to investigate how a Danish software organization integrates UX in their (self-labelled) agile development. While previous research has used Conboy (2009) to distinguish how a practice may affect the agility of either project or firm (Persson et al. 2016), we use it here to distinguish between development and

UX. Limited research has focused on how particular practices for integrating UX into agile development specifically contribute to agility pertaining to development and UX. We contribute to this research with empirical knowledge through a case study (Lazar et al. 2017) of integration practices in a Danish software company that prominently pursue both UX and agile development. Against this backdrop, we address the research question:

How do particular practices of integrating UX activities with agile development contribute to agility?

Agility and UX

Agility of software development and agile methods have played a significant role for almost two decades. This emphasis of agility has been explicit since the Agile Manifesto appeared in 2001, but can be found in much earlier research and practice. Several development methods are agile, e.g., Scrum, Extreme Programming, and Lean are well known, but there are also strong agile elements in Unified Process and in many other methods and practices. The literature on agile software development methods was initially ignorant of the literature on organizational agility (Mathiassen & Pries-Heje 2006) and remains almost singularly connected to software development (Larman 2004) entailing limited attention to UX.

Several studies have compared the agile methods and the elements suggested in an agile method, such as; Scrum or extreme programming, to the actual practices of the software developers. Many practices are tailored to the specifics in the development companies, situations, and conditions, e.g., having a customer on site in the development room gets neglected or tailored

Conboy's (2009) theory of agility

1. A process component must contribute to at least one of:
 - (i) creation of change
 - (ii) proaction in advance of change
 - (iii) reaction to change
 - (iv) learning from change
2. A process component must contribute to at least one of, and must not detract from:
 - (i) perceived economy
 - (ii) perceived quality
 - (iii) perceived simplicity
3. A process component must be continually ready, i.e., minimal time and cost to prepare the component for use.

due to the customer's reluctance to commit the necessary effort or simply the absence of an identifiable customer (Hoda et al. 2011; Wang et al. 2012).

Agile methods lack of focus on system usability, usability work, and UX work has witnessed a growing research interest (Silva da Silva et al. 2011; Wale-Kolade et al. 2013; Salah et al. 2014). A review (Wale-Kolade et al. 2013) found 7 claims across the research literature (see sidebar "Claims ..."). These claims do not come without criticism as, for example, claim 6 is supposed to ensure that user experience concerns are always present, but a study shows that UX designers may well identify too closely with the development goals and lose track of the UX concerns (Detweiler 2007). There are many other rebuttals raised against these seven claims. From this it is concluded that the integration of UX practices in to agile development is not simple and that matching several concerns and situation will be necessary (Wale-Kolade et al. 2013).

Agile development, first suggested as an ideal for software development, has gradually rose to being important for organisations in general, and for UX practice. In search for a theory of agility that is formulated in terms of development in general by relying here on the theory by Conboy. Conboy proposes the following definition of agility of development method agility emphasizing the core principles of agility of embracing change and providing customer value: *"[T]he continual readiness of an ... method to rapidly or inherently create change, proactively or reactively embrace change and learn from change while contributing to perceived customer value (economy, quality and simplicity) through its collective components and relationships with its environment"*

(Conboy 2009, p. 340). The definition is translated into a formative taxonomy of agility (see sidebar "Conboy's theory of agility"). While previous research has used (Conboy 2009) to distinguish how a practice may affect the agility (Persson et al. 2016), we use it here to distinguish between development and UX.

Research approach

Our investigation of integrating UX with agile development employed a single case study approach (Lazar et al. 2017) to address the research question on how specific practices contribute to agility. The case setting is a Danish software company with 100 employees that prominently pursue both UX and agile development. Most employees have a master's degree related to user experience, interaction design or software development. The company develops software products for various domains as ordered by their customers. The customer base consists of small and large private and public organizations. Sometimes employees assist with specialist knowledge on time-and-material contracts in customer organizations for up to several months. The company was recently awarded the prize as "IT comet of the year" in Denmark, partly because UX is highly emphasized in product development. Ten percent of the employees are UX consultants. UX activities are also considered well integrated within the case company as the UX consultants play a central and managerial role in most projects. We interviewed 10 employees with roles such as UX consultant, software developer, project manager, and software architect. We conducted a theory directed content analysis of the case data using Conboy's (2009) theory of agility (see sidebar) attending to how the effect of integration practices differs for the software developers and UX consultants.

Findings

The case company has adopted two process elements that are primary for integrating UX with agile development: Upfront-design and work-in-parallel.

The objective of upfront-design is to consider the software product from various perspectives before producing any code. The UX consultant is the main actor but collaborates with the system architect (an experienced software developer), the customer, and the users. Collaboration with the architect is considered critical in order to reduce development risks. In this phase the UX consultants typically participate full time having responsibilities of defining system requirements, making design sketches, and designing wireframes. Wireframes are validated in collaboration with the system architect and customers. System requirements linked to wireframes are considered the primary outcome of the upfront-design.

The work-in-parallel primarily entails software developers producing code based on the specifications elicited in the upfront-design. Here the UX consultants spend around 20% of their time as they are also allocated to work on other projects. The work-in-parallel starts by establishing the development team and involves a series of sprints lasting 3-4 weeks. The team typically includes one UX consultant, 2 to 4 developers, and a project manager. The main responsibility of the UX consultant is to review and sign off on implemented designs at the end of each sprint and facilitate product evaluations with customers. The UX consultant is also located in close physical proximity to the developers, aiming to foster easy access. Wireframes function as a central communication medium for interfacing between the upfront-design and work-in-parallel. UX consultants use wireframes when

they meet the remainder of the development team to convey system requirements.

Tables 1 and 2 outline how the upfront-design and work-in-parallel contribute to agility from the perspective of the developers and UX consultants. During upfront-design (table 1), the architect contributes to agility by being proactive to change in the architectural design of the system by anticipating development risks. This is possible due to coordination with the UX consultant who readily reacts to specification changes from the very beginning and learns from these changes, e.g. through heavy use of design iterations in dialogue with customers and users. The UX consultants' responsibility to identify requirements by coordinating with the customer and users is a separation of concerns that gives simplicity for the architect who only needs to coordinate with the UX consultant. In addition, the UX consultant improves product quality through strong interaction with customers and users. At this early stage, changes in requirements and designs lead to improvement from an economic perspective.

Work-in-parallel (table 2) involves the UX consultant meeting with the development team at the end of a sprint to review implemented designs. This limited time scope of the reviews makes reactions to change more manageable for the software developers as they do not have to cope with changes continuously. On the other hand, this enables the UX consultant to focus on creating change from the customer perspective. The less and more formalized UX input also contributes to agility by inducing simplicity from the developers' perspective while improving economy of scale for the UX consultant, who is able to work on multiple projects.

Agility (Conboy 2009)	Developers	UX consultants
1. A process component must contribute to at least one of: <i>(i) creation of change</i> <i>(ii) proaction in advance of change</i> <i>(iii) reaction to change</i> <i>(iv) learning from change</i>	Prior to development, the architect is <i>proactive to changes</i> and anticipate development risks	Initial <i>reaction to change and learning from change</i> is high with strong customer dialog and no software committed
2. A process component must contribute to at least one of, and must not detract from: <i>(i) perceived economy</i> <i>(ii) perceived quality</i> <i>(iii) perceived simplicity</i>	Improved <i>simplicity</i> by separation of concern and less coordination with users	Improved <i>quality</i> by intense interaction with users and customers and improved <i>economy</i> for initial changes
3. A process component must be continually ready i.e. minimal time and cost to prepare the component for use.	Architect is available and active	UX consultant autonomously control the project scope

Table 1: Contributions to agility by upfront-design

Agility (Conboy 2009)	Developers	UX consultants
1. A process component must contribute to at least one of: <i>(i) creation of change</i> <i>(ii) proaction in advance of change</i> <i>(iii) reaction to change</i> <i>(iv) learning from change</i>	UX feedback and design changes limited to sprint reviews makes <i>reaction to change</i> more manageable	Thorough reviews and quality control detached from the development team while in dialog with customers help <i>creation of change</i>
2. A process component must contribute to at least one of, and must not detract from: <i>(i) perceived economy</i> <i>(ii) perceived quality</i> <i>(iii) perceived simplicity</i>	Improved <i>simplicity</i> by less and more formalized input on UX	Improved <i>economy</i> of scale by working on multiple projects
3. A process component must be continually ready, i.e. minimal time and cost to prepare the component for use.	Stable and homogeneous developer teams	Reuse of UX competences and insights across projects

Table 2: Contributions to agility by work-in-parallel

Conclusion

This paper reported how the integration of UX activities with agile development by upfront-design and work-in-parallel may contribute to agility. While previous research has reported on both practices (Wale-Kolade et al. 2013), we present a novel elaboration based on a case study in a Danish software company and Conboy's (2009) theory of agility. We show how the two practices contribute substantially different to agility for developers and UX consultants in their dealings with change, simplicity, quality, economy, and readiness. Our findings show an opportunity for deepening the discussion among researcher and practitioners on what agility entails for software development and UX in practice.

References

- Conboy, K., (2009). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20(3):29–54.
- Detweiler, M., (2007). Managing UCD within agile projects. *Interactions*, 14(3):40-42.
- Larman, C., (2004). *Agile and Iterative Development: A Manager's Guide*, Addison-Wesley Professional.
- Lazar, J., J. H. Feng, and H. Hochheiser. 2017. *Research Methods in Human-Computer Interaction*. 2nd ed., Cambridge, Morgan Kaufmann.
- Mathiassen, L., and J. Pries-Heje, (2006). Business Agility and Diffusion of Information Technology. *European Journal of Information Systems*, 15(2):116-119.
- Persson, J. S., J. Nørbjerg, and P. A. Nielsen, (2016). Improving ISD Agility in Fast-Moving Software Organizations. In Proceedings of ECIS, 2016.
- Salah, D., Paige, R. F., & Cairns, P. (2014). A systematic literature review for agile development processes and user centred design integration. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ACM.
- Silva Da Silva, T., M. S. Silveira, C. de O. Melo, and L. C. Parzianello, (2013). Understanding the UX designer's role within agile teams. In International Conference of Design, User Experience, and Usability, Springer.
- Wale-Kolade, A. P. A. Nielsen, and T. Päiväranta, (2013). Usability work in agile systems development practice: a systematic review. In *Building Sustainable Information Systems*. Springer.
- Wale-Kolade, A. and P. A. Nielsen, (2016). Apathy towards the integration of usability work: a case of system justification. *Interacting with Computers*, 28(4):437-450.
- Wang, X., K. Conboy, and M. Pikkarainen, (2012). Assimilation of Agile Practices in Use. *Information Systems Journal*, 22(6):435-455.