Information and Software Technology 102 (2018) 1–5

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

The evolution of agile UXD

Tiago Silva Da Silva[a,*], Milene Selbach Silveira[b], Frank Maurer[c], Fábio Fagundes Silveira[a]

[a] Universidade Federal de São Paulo, Brazil
[b] Pontifícia Universidade Católica do Rio Grande do Sul, Brazil
[c] University of Calgary, Canada

A B S T R A C T

Context: Agile User eXperience Design (Agile UXD) is a current theme and a trending topic for the future of software development. The integration of UX Design within Agile development is seen as one of the frontiers for Agile Methods as a balance between upfront design as advocated by UX and the you-ain't-gonna-need-it (YAGNI) principle from the agile community must be found.
Objective: In this paper, we analyze the evolution and current state of Agile UXD to provide a brief overview of the topic and to point out still unaddressed gaps, challenges, and future trends.
Method: We systematically analyzed the existing research literature on how this topic evolved over time. We identified three categories with distinctive sets of work and classified them as Early, Middle and Recent years.
Results: We noticed that the Process and Practice dimension has already crossed the line that separates Agile and UXD, the People and Social dimension is crossing this line right now, and the Technology and Artifact is the dimension that took the longest to be addressed, and it did not cross the line yet. Crossing the line means that there is already a full understanding from the Agile side of UX needs and vice versa.
Conclusion: Agile UXD is a need for today's software development teams. However, integrated teams still need to understand that UXD is not a role, but discipline and culture for the whole Agile environment.

1. Introduction

In 2011, during the Agile Manifesto's 10th Anniversary Reunion at the Agile Conference held in Salt Lake City, one of the questions asked was ``What is the next frontier for Agile?", to which Martin Fowler answered:

> "There are two… the integration of operations and the integration with User eXperience (UX) work… I remember, not many years ago, UX people saying: you could not possibly do Agile UX. Everything has to be planned in advance".
>
> (Agile [1]).

Nowadays, we are living in an experience-driven world: the User Experience (UX) of a software product often determines its success or failure, especially when it faces an end-user market. Therefore, if you are in the software business, you are probably in the UX business (Gothelf and Sneiden, [10]). Also, Agile software development has been characterized differently than plan-based or traditional development methods, mainly with the focus adapting to change and delivering products of high quality through simple work-processes [15].

Henceforth, we are going to address the use of User eXperience Design (UXD) approaches within Agile processes as Agile UXD.

While Agile focuses on the question of how useful software can be developed, User eXperience Design (UXD) ensures that the goals and needs of the end users are the focus of the development of a product. There is an inherent tension between both schools of thought: agile approaches usually try to reduce and limit upfront analysis and design work while UXD approaches emphasize the need for these. This tension is a core reason why researchers, seeing the value of both arguments, have been investigating how to integrate both approaches.

As delivering highly usable systems is crucial for economic success and for creating business value in a fast-changing environment, the integration of Agile and UXD has been seen as a promising endeavor and has received increasing attention in the last 15 years.

In this paper, we analyze the evolution of Agile UXD – from an academic perspective, examining the literature published in peer-reviewed conferences and journals, and a recent published paper collection [7] on the topic –to provide a brief overview on the topic. We will also highlight still unaddressed gaps, challenges, and future trends. The primary objective here is not to present an extensive analysis of the literature, but, based on a comprehensive study of existing work, provide an overview to students, researchers, and practitioners who intend

* Corresponding author.
E-mail address: silvadasilva@unifesp.br (T.S. Da Silva).

https://doi.org/10.1016/j.infsof.2018.04.008
Received 21 February 2017; Received in revised form 19 March 2018; Accepted 18 April 2018
Available online 01 May 2018
0950-5849/

to better understand the evolution of Agile UXD.

2. Research method

To achieve the goals of this paper, we followed the strategy adopted by Brhel et al. [14]. We cross-checked their paper set with our results using the same strings and same databases to identify new milestones from 2012 to 2016. We analyzed the studies that focused on the subject of agile user experience design and classified them along dimensions as described in the next section.

Brhel et al. [14] carried out a literature review following the established guidelines for conducting systematic reviews suggested by Kitchenham [3]. These are a proven means to arrive at a complete and thorough overview of existing research within a domain. We followed their strategy because, to the best of our knowledge, the study presented by Brhel et al. is the latest and most complete literature review on the topic.

Due to our focus – peer reviewed literature – we did not include books like Beyer [8]; Ratcliff and McNeill [12]; Brown [4]; Jongerius et al. [16]; Klein [11]; and Gothelff and Snieden [10] in our study. However, we acknowledge that some insights are condensed into these textbooks.

The contents of the Cockton et al. [7] book is included in our review because the book is essentially a collection of peer reviewed papers from a NordiCHI workshop in 2014 entitled 'On the integration of user centred design in agile development' (Larustottir et al., [13]). Thus, instead of referencing the papers from the workshop, we refer to the extended and improved versions in the book.

To illustrate the evolution within phases, we defined the phases using intervals of five years for each phase to distribute the stages equally. The milestones were chosen based on the first appearance of that topic in a paper. For instance, we do not mention every framework proposed, we just highlight the first one considering the chronological order in the timeline. The reader can find all the framework proposals in the reference list.

3. The evolution of agile UXD over time

Based on the work of Barksdale and McCrickard [9], Brhel et al. [14] defined four dimensions in which the existing literature about the theme is classified: process integration, practice integration, people and social integration, and technology integration. We expanded this classification by adding artifact integration to the technology dimension. Artifact integration represents the incorporation and adaptation of artifacts from both UXD and Agile to mediate teams' communication.

We also merged the processes and practices integration dimensions as well because we consider these aspects strongly related and almost inseparable. As a result, we are using the following dimensions in our analysis:

— *Process and Practice* integration is understood as the merging and synchronizing of UXD and Agile processes, providing a unified process incorporating both perspectives as well as embedding of UXD practices into Agile processes and vice versa.
— *People and Social* integration means changes to the team composition to bring experts from the two different disciplines together as well as the social interaction and the joint creation of knowledge.
— *Technology and Artifact* integration entails the use of technological means to support and coordinate activities as well as the incorporation and combination of artifacts from both processes to mediate communication and create a shared understanding of issues.

Both Figs. 1 and 2 outline the evolution of the field by partitioning it in three periods – Early, Middle, and Recent years. In Fig. 1, we emphasize the first publication addressing a specific topic. This chart illustrates the topical evolution of the field.

In contrast, Fig. 2 highlights milestones in the evolution by linking topics with papers. In Fig. 1, each dimension is presented for each period in more specific themes, each of which is actually addressed by a paper that is referenced in Fig. 2. In Fig. 2, the milestones – the first time a subject is addressed in a publication – for each dimension in each period are presented.

The timeline in Fig. 2 shows the Agile UXD evolution divided into three Periods – with intervals of five years between each phase: Early, Middle and Recent. In the Early years, the discussion of a topic took started with speculative studies on the its importance. During the Middle period, Agile UXD began to identify its particularities in relation to other fields and established its own identity. During the recent years, the community pushed the boundaries and started to face some new limitations to overcome.

4. What IS next for agile uxd?

Although Agile UXD is an established research topic, there is always work to be done, issues to be addressed and discoveries to be made.

The overall evidence that we should not manage and control two separate processes is solid. However, we still have a long way to go. Most people admit that while the integration of agile and UXD processes is not always smooth, it is a step forward compared to each of them individually. The adoption of agile and UXD approaches in the industry has grown steadily.

However, there are still open questions and unaddressed gaps.

Currently, there is still a need for new framework proposals to synchronize usability evaluations (UXD perspective) with unit testing or acceptance testing (Agile perspective). Although there are some framework proposals reported in the literature – e.g., Mostafa [6] – agile methods are continually evolving, which requires new ways to integrate UXD concepts. While agile and UXD methods have been combined in several environments, it is essential to develop clear integration guidelines and empirically validate them.

The daily operation of an Agile UXD process is still a concern for developers and designers. They understand the importance of each other's work but still do not know how to make it work on a day-to-day basis. For instance, according to Version One [17], when Agile professionals are asked how to measure progress on a daily basis, customer and/or user satisfaction was only the 7th metric cited by the respondents – behind velocity and iteration, and release burndown.

UXD work tends to be distributed throughout the entire development process, requiring continuous research, continuous design, and continuous evaluation. This implies a need to share the results of UX work with the whole team on an ongoing basis, allowing the team to build a shared understanding. While there are some recent suggestions in the literature to deal with creating a shared understanding between UX designers and developers in agile teams, there are still concerns. The problem of combining UXD and agile methods is an example of a context-dependent issue. Different teams in different contexts use different artifacts and techniques to create a shared understanding.

Another concern is related to the organizational culture. User-Centered Design must mesh with the Agile organizational culture in such a way that everyone in the team will understand UXD as a team discipline rather than a role in the team. A solid understanding of Agile and UXD cultures and practices can help both, developers and designers, to adjust their methodologies, and to adopt techniques that would improve their lines of communication.

This cultural change leads us to face another concern: the future of UXD professionals inside the organization. UXD Specialists have increasingly been working as business analysts as well as coaching development teams to familiarize them with the UX culture.

Distributed teams are a reality and will be increasingly common. Teams will be able to work smoothly with respect to the UX of a software product only with the integration of UXD into the team and the

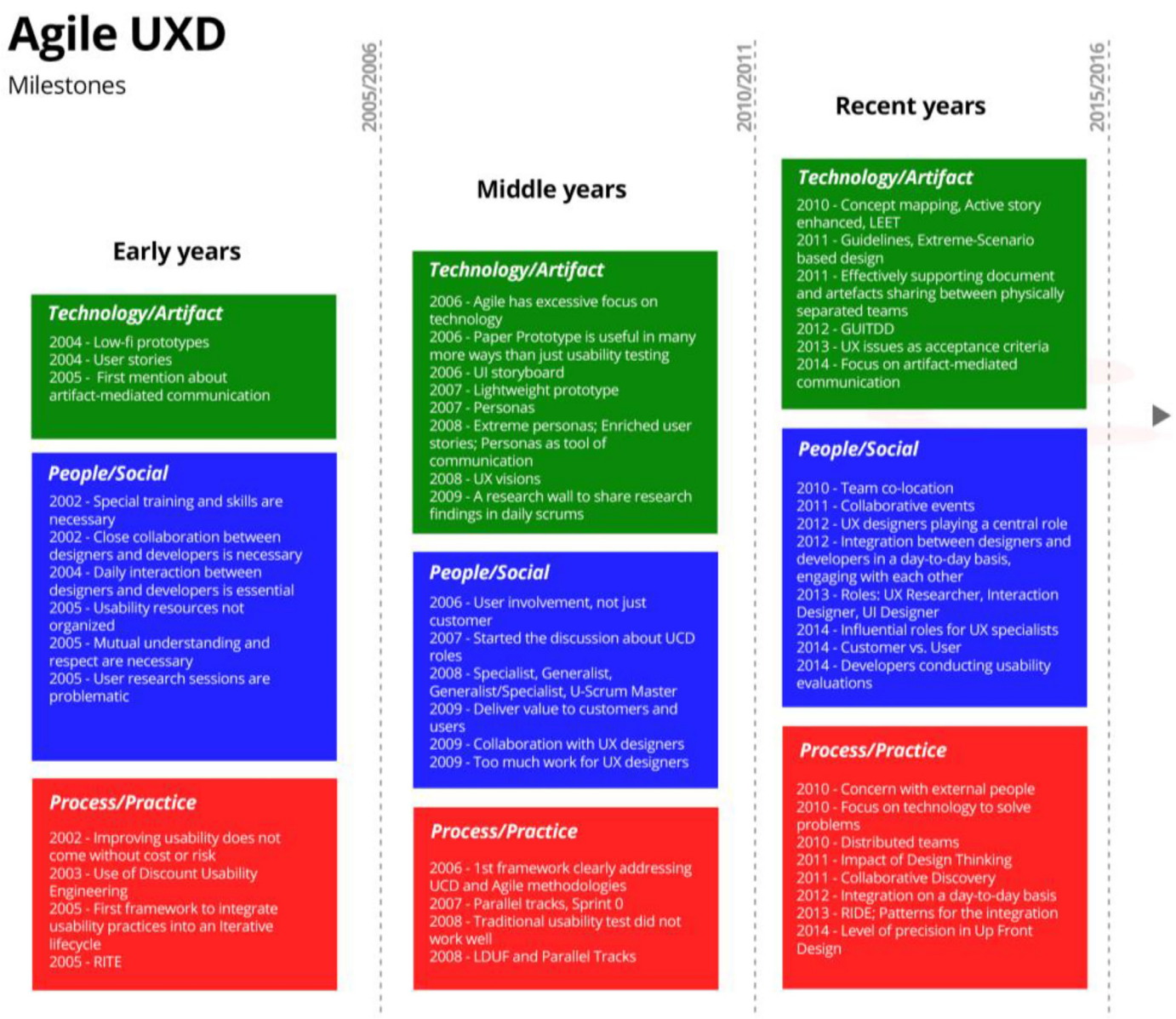

Fig. 1. Agile UXD over time and its milestones.

company culture. UXDD (User eXperience Driven Development) [5] may become more widely used. The main idea of UXDD is that, before you get into coding mode, you have customers sign off on wireframes and storyboards for each task offered through the presentation. Looking back 15 years ago, we would only test software if there was enough time at the end of the process. Today, 34% of agile teams use test-driven development, moving test-automation to the forefront of the development process (Version One, [17]).

With a possible UXDD, we need to be innovative in building tools that enable us to perform this user experience driven development. InVision & Marvel are moving in this direction. Another gap is the use of software analytics approaches both to gather requirements and to analyze usage data for making improvements. HotJar is an example of this approach.

Due to distributed teams, there is already a need for digital artifact-mediated communication. This necessity takes us to the need for tools that enable the integration of UI development, as there is in the context of continuous integration. Analyzing the report concerning tools provided by Version One [17], the ``customer idea management tool" is the least used one. Why is this the case? For a tool to be seen as useful, it is necessary that the teams should see value in using them and find them suitable to their work practices. For instance, relying on TDD would not be possible without a good support tool.

The development of computer-assisted usability engineering tools has been discussed for quite some time and there is still no agreement on which the best tools are. Teams tend to appropriate existing tools and choose tools that fit their circumstances.

Tools are needed to support developers in acquiring and sharing UXD and software engineering best practices. They should also be flexible enough for developers to fit them into their particular project context. As said by Seffah and Metzker [2], Agile UXD will be considered more seriously at large if and only if a computer-assisted usability engineering platform is available.

In a long-term, we believe that Agile UXD will be core to the software development culture just as Agile is today. Both processes will be fully integrated.

Agile UXD will be a standard followed by teams that develop interactive software so that they will develop bearing users in mind. There will be infrastructure available for the integration of Agile UXD with operations, which will allow continuous delivery of positive

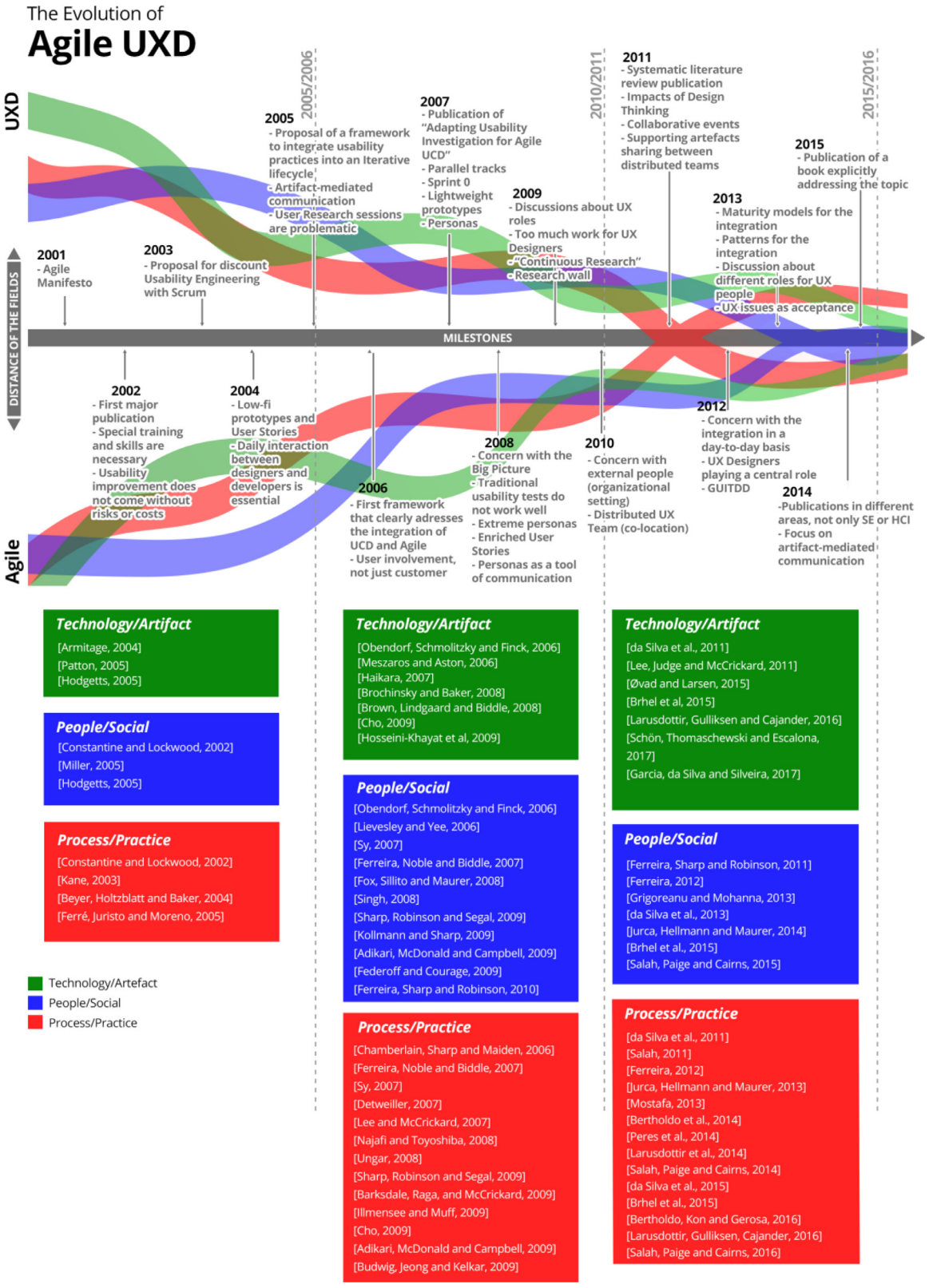

Fig. 2. Agile UXD over time, its milestones and references.

experiences to the end user.

Perhaps, we will be able to predict expected user experiences based on data from previous experiences of groups of users – for example, data from a group of people that represents a particular persona – with the interactive computing system.

5. Final remarks

Work on the Process & Practices dimension (Red Line in Fig. 2) already began in 2002 – based on our conceptual drawing presented in Fig. 2. However, in the beginning, it was just a union of practices from

both approaches without any adaptation. Nowadays, Process and Practice have crossed the line that separates Agile and UXD since the understanding that we cannot have two separate processes is clear.

In 2002, UXD and Agile were far apart regarding the People and Social dimension (Blue line) and now, due to the focus on people and cultural changes, we believe that teams are crossing the line that separates the two fields.

Finally, Technology and Artifact (Green line) is the dimension that took the longest to be addressed. Commercial grade tools that address this dimension are still missing and there is still a way to go before we can achieve an integration.

Lastly, based on an understanding and extensive analysis of the academic literature published so far, we presented an overview of the field and we argue that the topic has reached such a maturity that discussions about its importance are no longer necessary. The goal of the Next stage – to where we are going after the Recent years – then becomes not to create a legacy as much as to simply make sure that the legacy lasts.

If we really want to make Agile User eXperience Design cross the line entirely, we need to really understand users and they must not only be well represented, but also be a real part of the process, regardless of context.

In conclusion, we believe that this short paper may benefit academics in a broader way, due to its wide landscape, as well as students who think that just XP, Scrum, and/or Kanban are enough for all their problems, and industrial readers who will find several references that represent the current body of knowledge.

Acknowledgments

The authors would like to thank FAPESP and the anonymous reviewers whose feedback immensely helped to improve the paper.

References

[1] A. Alliance, Event: The Agile Manifesto 10th Anniversary Reunion, (2011) August 2011)Retrieved July 10, 2015 from http://www.agilealliance.org/resources/learning-center/event-the-agile-manifesto-10th-anniversary-reunion.

[2] A. Seffah, E. Metzker, The obstacles and myths of usability and software engineering, Commun. ACM 47 (12 December 2004) (2004) 71–76 DOI= http://dx.doi.org/10.1145/1035134.1035136.

[3] B.A. Kitchenham, *Guidelines for* Performing Systematic Literature Reviews *in* Software *Engineering*, Keele University, Keele, UK, 2007 *EBSE* Technical Report *EBSE-2007-01*2007.

[4] D.D. Brown, Agile User Experience Design: A Practitioner's Guide to Making It Work 1 Morgan Kauffmann, 2013, p. 256.

[5] D. Esposito, Modern Web Development: Understanding Domains, Technologies, and User Experience 1 Microsoft Press, 2016, p. 448.

[6] D. Mostafa, Maturity Models in the Context of Integrating Agile Development Processes and User Centred Design, Ph.D thesis University of York, 2013.

[7] G. Cockton, M. Lárusdóttir, P. Gregory, Å. Cajander, Integrating User-Centred Design in Agile Development, Human-Computer Interaction Series 1 Springer International Publishing, 2016, p. 276.

[8] H. Beyer, User-centered agile methods, Synthesis Lectures on Human-Centered Informatics 1 Morgan & Claypool Publishers, 2010, p. 80.

[9] J.T. Barksdale, D.S. McCrickard, Software product innovation in agile usability teams: an analytical framework of social capital, network governance, and usability knowledge management, Int. J. Agile Extreme Softw. Dev. 1 (1 June 2012) (2012) 152–177.

[10] J. Gothelff, J. Snieden, Lean UX: Designing Great Products with Agile Teams, second ed., O'Reilly Media, 2016.

[11] L. Klein, UX For Lean Startups: Faster, Smarter User Experience Research and Design 1 O'Reilly Media, 2013, p. 240.

[12] L. Ratcliff, M. McNeill, Agile Experience Design: A Digital Designer's Guide to Agile, Lean, and Continuous (Voices That Matter). 1 New Riders, 2011, p. 320.

[13] M. Larusdottir, Å. Cajander, J. Gulliksen, G. Cockton, P. Gregory, D. Salah, On the integration of user centred design in agile development, *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational* (NordiCHI '14), New York, NY, USA, ACM, 2014, pp. 817–820 https://doi.org/10.1145/2639189.2654836.

[14] M. Brhel, H. Meth, A. Maedche, K. Werder, Exploring principles of user- centered agile software development: A literature review, Inf. Softw. Technol. 61, 1 (May 2015) (2015) 163–181.

[15] T. Dingsøyr, T. Dybå, N.B. Moe, Agile Software Development: Current Research and Future Directions 1 Springer-Verlag, Berlin Heidelberg, 2010, p. 238.

[16] P. Jongerius, A. Offermans, A. Vanhoucke, P. Sanwikarja, J. van Geel, Get Agile!: Scrum for UX, Design & Development 1 BIS Publishers, 2013, p. 144.

[17] Version One. 2015. State of Agile Survey 2014. Agile Made Easier. Version One. 116; pages.

Tiago Silva da Silva received his M.Sc. from Pontifical Catholic University of Rio Grande do Sul in 2008 as well as his Ph.D. in Computer Science in 2012. He was a visiting researcher at University of Calgary in 2010 and 2011. He is currently an Assistant Professor at the Science and Technology Institute of Federal University of São Paulo (UNIFESP), doing research on agile software development, user-centered design, integration of agile and user-centered design, software fault and usability prediction, and evidence-based software engineering.

Milene Silveira received her M.Sc. in Computer Science from Federal University of Rio Grande do Sul in 1996 and her Ph.D. in the same area from Pontifical Catholic University of Rio de Janeiro in 2002. She is a Professor at Pontifical Catholic University of Rio Grande do Sul since 1994, doing research on human-computer interaction, focusing on its relations with agile software development, data visualization, end-user development, and social networks.

Frank Maurer is a professor of computer science. He was one of the first researchers investigating agile methods and is widely published in the area. His current research interests are Agile methods, multi-surface systems, and engineering immersive analytics applications. He was program chair and program committee member of a substantial number of agile conferences over the last 15 years.

Fábio Silveira received his M.Sc. from Federal University of Rio Grande do Sul in 2001 and his Ph.D. in Computer Science from Technological Institute of Aeronautics (ITA) in 2007. He was a visiting researcher at Technische Universität Berlin from June to August 2009. He is currently an Associate Professor at Science and Technology Institute of Federal University of São Paulo (UNIFESP), doing research on object-oriented and aspect-oriented software testing, experimental software engineering, agile methods, and metadata.