

A Methodology for Integrating User Experience Methods and Techniques into Agile Software Development

Andreas Hinderks¹[0000-0003-3456-9273]

supervised by Francisco José Domínguez Mayo and Jörg Thomaschewski

¹ University of Seville, Spain
andreas.hinderks@iwt2.org

Abstract. Agile methods are used more and more frequently to develop products to reduce development time. However, for developing desirable products with good user experience, Agile methods alone are not enough. This Ph.D. thesis explores how an iterative and incremental process model in the context of Agile methods must be designed in order to manage and improve product quality in terms of user experience. For this purpose, the design science research methodology is used. Using existing and new methods, an iterative and incremental process model is developed, which is validated in an international study. This Ph.D. thesis contributes to the knowledge base of software development by providing a process model that offers an iterative and incremental improvement of product quality in terms of user experience and process quality.

Keywords: agile software development, agile methodology, human-centred design, human-computer interaction, product quality, process improvement

1 Introduction

Software development companies use agile methodologies to develop products more efficiently. Agile methodologies (e.g. Scrum [24], Kanban [1], or Extreme Programming (XP) [2]) reduce the time taken to develop a product available to the market [25]. The iterative approach to developing software minimises the risk of developing software that is not in line with the market [3]. By performing retrospectives [24] at the end of an iteration, both the product quality and the agile process quality can be improved.

Nowadays, it is no longer enough to be faster on the market—the product is also expected to have a high level of user experience (UX). Users expect to be able to use the product to solve their tasks in a quick and efficient manner without having to put in any major effort. Furthermore, for a product to succeed, it is essential also to consider a good UX.

A well-known definition of UX is given in ISO 9241-210 [9]. Here UX is defined as “a person’s perceptions and responses that result from the use or anticipated use of a product, system or service”. Thus, UX is seen as a holistic concept that includes all types of emotional, cognitive or physical reactions concerning the concrete or even

only the assumed usage of a product formed before, during and after use. A different interpretation is to define UX as a set of distinct quality criteria [19] that includes classical usability criteria, like efficiency, controllability or learnability, and non-goal directed or hedonic quality criteria [6], like stimulation, fun-of-use, novelty, emotions [17], or aesthetics [28]. This has the advantage that it splits the general notion of UX into a number of simple quality criteria (UX factors), which describe distinct and relatively well-defined aspects of UX that can be measured independently.

Agile methods themselves focus on developing useful products and not on developing products with a good UX. Development approaches involving Human-Centred Design (HCD) [9] are used to develop products with a good UX. These are, for example, *Design Thinking* [16], *Design Sprint* [13], etc. Additionally, processes have been developed to put HCD into practice [10]. But these HCD processes are isolated from the Agile methods themselves. Rather, it can be assumed that HCD and Agile methods are not integrated, as described in Section 2.

This article gives an overview of the research objective of my Ph.D. thesis and presents the actual progress. A first draft of an iterative and incremental process model for Agile methodologies is proposed, including new methods and artefacts. The paper is structured as follows: Section 2 briefly summarises the related work. Section 3 presents the research objectives and research questions of this Ph.D. thesis. Section 4 outlines the research methodology, covering a description of each step of the research process. Section 5 presents the first draft of an iterative and incremental process model for Agile methodologies. The paper ends with Section 6, with conclusions and ideas for future work.

2 Related work

In literature, many reviews have been conducted to research regarding the integration of Agile methods and User Centred Design (UCD) or Human-Centred Design (HCD). In the following paragraphs, the most important ones are briefly summarised in terms of their chronology.

Silva et al. [26] conducted an SLR on the integration of Agile methods and UCD in 2011. The authors analyse how usability problems are handled in agile projects and identify the following key aspects that play an essential role in integration: *little upfront design, prototyping, user stories, user testing, inspection evaluation, and one sprint ahead*.

In 2014, Salah et al. [21] analysed a similar area with their SLR, similar to Silva et al. The analysis of the authors identify the factors for the integration of Agile methods and UCD. Moreover, the authors examine the challenges and key aspects to ensure successful integration. The identified key aspects are *lack of allocated time for upfront activities, difficulty of modularization, optimizing the work between developers and UCD practitioners, performing usability testing, and lack of documentation*.

Brhel et al. [4] published in 2015 an SLR capturing the principles of user-centred agile software development (UCASD). The authors aimed to assess the current state of the art regarding the integration of agile methods and UCD. Using a coding system,

the authors extracted five principles: *separate product discovery and product creation, iterative and incremental design and development, parallel interwoven creation tracks, continuous stakeholder involvement, and artefact-mediated communication.*

In a study published in 2018, Silva et al. [5] analyse the results reported by Brhel's SLR [4] concerning the current status of the integration of Agile methods and User Experience Design (UXD). The outcomes of the respective publications are divided into three dimensions: *Process and Practice, People and Social, and Technology and Artefacts.* The individual outcomes are arranged on a timeline so that the chronological sequence of the publications can be visualised. The authors state in their analysis that solutions concerning integration are already being offered for the dimensions *Process and Practice* and *People and Social.* Finally, the authors conclude that solutions for *Technology and Artefacts* are still needed in order to achieve integration between Agile methods and User Experience Design to Agile UXD.

The overall impression from the related work is that most of the problems regarding the integration of Agile methods and UCD have been addressed. SLRs have adequately described the problem area in recent years. However, not all issues regarding integration have been solved. In the *Technology and Artefacts* dimension, there are still large gaps to be closed. *Technology* means to support and coordinate activities within the development process. On the other hand, *Artefacts* from UCD and Agile methods are understood as a means to communicate information or knowledge in the development process. This Ph.D. thesis aims to contribute to the closure of these gaps.

3 Objectives and research questions

The primary research objective of this Ph.D. thesis is to investigate how quality in terms of UX can be managed and improved in Agile methods. For this purpose, an iterative and incremental process model is to be developed that can significantly support the improvement of the UX. The process model is used in the context of Agile methodologies and should be integrated into existing agile process models. To this end, the following research questions (RQ) guide the research:

- RQ1: What are the problems by using existing technologies and artefacts supporting the integration of UCD and Agile methods?
- RQ2: What requirements must an iterative and incremental process model fulfil to support the improvement of the product quality and process quality in the context of the integration of UCD and Agile methods?
- RQ3: How should an iterative and incremental process model be designed to meet the requirements of RQ2?
- RQ4: How can the developed process model be applied in the real world?
- RQ5: How can the developed process model be evaluated?

The classification system given by Wieringa [31] can be used to classify the research questions. RQ1 and RQ5 can be classified as knowledge questions. These questions aim to identify the problems to integrate UX methods and techniques into Agile methods (RQ1), and to determine whether the proposed process model is adequate for

solve these problems (RQ5). On the other hand, RQ2, RQ3, and RQ4 can be classified as practical problems. These research questions focus on developing and applying the process model itself.

4 Research methodology and progress

This Ph.D. thesis follows the Design Science (DS) research methodology provided by Johannesson and Perjons [11] in combination with the guidelines by Wieringa [31] and Peffers et al. [18]. **Fig. 1** gives an overview of the process model, which includes the different activities performed in this Ph.D. thesis.

Activity 1: Explicate Problem	Activity 2: Define Requirements	Activity 3: Design and Develop Artefact	Activity 4: Demonstrate Artefact	Activity 5: Evaluate Artefact
What is the problem experienced by some stakeholders of a practice and why is it important?	Define requirements for process model Create a vision of the outcome	Create process model Evaluate against predefined requirements	Build real world application of process model	Conduct studies to validate process model

Fig. 1. Design Science research activities and process

In the next paragraphs, the current status of each activity from **Fig. 1** is described in more detail. Additionally, an evaluation strategy is given for each resulting artefact from the research questions.

4.1 Explicate problem (related to RQ1)

The author has gained practical experience in the implementation of agile software projects with Scrum and Kanban in the past years. Thereby, the research and use of UX methods have always been of importance, especially the measurement of UX [7, 8, 23]. One research aspect is to identify techniques that could be used to determine the UX before actually developing the product increment. After some research, we come to the first conclusion that hardly any methods are used to determine the UX.

The UX can be evaluated after the product increment was developed. This is already a known and common practice. One requirement for software development, however, may be that the expected UX can be evaluated before development begins.

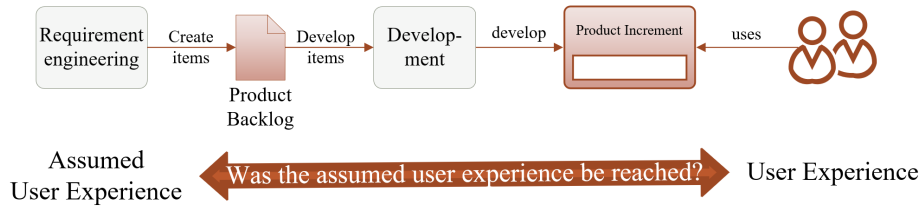


Fig. 2. Assumed User Experience and Actual User Experience

In agile software development methodology, all requirements from the Requirements Engineering (see **Fig. 2**, left side) are usually recorded in a product backlog. These are developed into a product increment by the development team after an evaluation (such as complexity and required time). The developed product increment is then used by the user (see **Fig. 2**, right side).

A product backlog item can or maybe should have an impact on the UX. After development, the UX can be measured using various methods [29]. Comparison with the previously assumed UX, however, is not possible, unless it has been systematically measured. Nowadays, the assumed influence on UX is not systematically measured. From our point of view, the resulting problem is as follows: There is no systematic verification between the assumed and the actual UX.

We are convinced that the problem and outcome of the Ph.D. thesis are essential and relevant for many companies worldwide. One reason for this is that processes, in general, can only be improved if they have been systematically recorded and evaluated beforehand. As stakeholders, we address all roles that are responsible for the product. These are, for example, the product owner [24] or the product manager. UX professionals such as UX researchers or UX designers are also addressed as stakeholders.

The Ph.D. thesis will be based on an SLR. The SLR aims to provide the current status regarding the integration of UX methods and techniques into agile software development. As a result, potential problems by using existing technologies and artefacts supporting the integration of UCD and agile methods will be addressed. The SLR answers the following RQs:

- RQ1.1: Which technologies and artefacts are used or supported in the integration of UCD and Agile methods?
- RQ1.2: What are potential problems by using founded technologies and artefacts found in RQ1.1?
- RQ1.3: How can the user experience in agile software development methodologies be estimated before development?
- RQ1.4: What proposals exist to determine the user experience for a product backlog item or a requirement?
- RQ1.5: What proposals exist to improve the user experience in the agile software development process?

Evaluation strategy: The SLR itself is the evaluation strategy to figure out potential problems. Furthermore, the guidelines given by Kitchenham and Charters [12] are adopted.

4.2 Define requirements (related to RQ2)

The iterative and incremental process model proposed in the Ph.D. thesis is to be applied to existing agile software development methodologies such as Scrum [24], Kanban [1] or Extreme Programming [2], and scaled Agile methods as well as LeSS [15] or SAFe [14]. The process model can be understood as a product. Each product, including the process model, should fulfill certain requirements for its users [30, 31], represented by the stakeholders (see Section 4.1). The following proposed requirements of the process model are to be evaluated as suggestions.

- The process model should consider the existing agile methodologies or be integrated into the agile methodologies.
- No additional roles should be created.
- Existing artefacts such as product backlog or user stories should be used. However, the basic structure of the artefacts should be preserved.

In addition to these requirements, certain structural qualities should also be considered:

- **Modularity:** The process model should be modular—i.e. either entirely or partly applicable.
- **Learnability:** The process model should make it possible to improve the process continuously.
- **Adaptability:** The process model should be easy to use and to integrate into existing agile software development projects.

Evaluation strategy: The requirements presented in this section still need to be evaluated with the stakeholders addressed in Section 4.1. Also, the list of requirements should be checked for completeness. Expert reviews should evaluate the requirements. If the results of the expert reviews are not clear, interviews or a Case study should be conducted.

4.3 Design and develop artefact (related to RQ3)

This phase mainly comprises of the designing and development of the artefact. The first task, which is completed, is done by creating the first draft of an iterative and incremental process model for agile development (see Section 5), which integrates UX in the agile development process. The second task, which is still ongoing, covers the iterative evaluation of the process model through expert reviews and stakeholders (see Section 4.1). We can learn how the process model can be tailored for different agile approaches (e.g. Scrum [24] or Kanban [1]) with focus on UX. The following task will be to develop the new methods of the process model. Existing methods must be named and integrated into the process model.

Evaluation strategy: The process model, including new and existing used methods, will be validated as a whole (see Section 4.5). The newly developed methods will be validated in a multi-level Delphi study.

4.4 Demonstrate artefact (related to RQ4)

The iterative and incremental process model is demonstrated by providing an application in a real-world context. The requirements, as defined in Section 4.2, can thus be checked in an initial validation. Any corrections can be made. The results can also be used to prepare the evaluation or validation described in Section 4.5.

Evaluation strategy: For this phase, quantitative evaluation methods such as laddering [20] are offered. Moreover, a combination of interviews and a questionnaire are also preferred.

4.5 Evaluate artefact (related to RQ5)

The evaluation of the artefacts from RQ1 to RQ4 was partially carried out individually and in isolation. The methods developed in the Ph.D. thesis and integrated into the process model are individually assessed. It is planned to use questionnaires, Delphi studies, and case studies.

Evaluation strategy: The process model is validated through an international case study. With this strategy, each method developed in this Ph.D. thesis and the process model itself with the individual methods are validated as a whole.

5 Proposed iterative and incremental process model for agile development

Based on the problem that there is no systematic verification between the assumed and the actual UX, (see Section 4.1), the first draft of a solution is developed. This draft represents a process model [27] for agile software development methodologies. This process model is based on the requirement given in Section 4.2. To be sure, the requirements have to be checked to see whether they are desired (see Section 4.2).

The process model consists of five well-known activities (see **Fig. 3**, no. 1 and 5–8) and four new activities (see **Fig. 3**, no. 2–4 and 9). In addition to the name of the activity, the resulting artefact and the method for validation are mentioned.

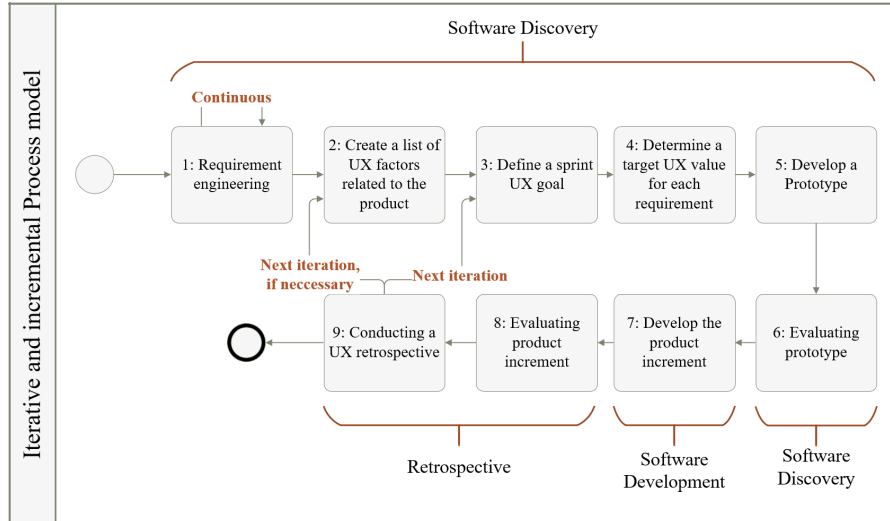


Fig. 3. Process Model Activities

The process model presented in this section integrates the plan-do-check-act (PDCA) principle in agile software development methodologies around the UX aspect. The process model aims to develop improvements through a continuous review of the achieved UX. The following list describes the individual activities from **Fig. 3**.

1. **Requirements engineering:** This continuous activity is well-known in software development. Several models and methods are used to apply requirement engineering in agile software development methodologies [22]. As a result of this activity, a list of requirements is created. In literature research, methods are collected that manage requirements.
2. **Create a list of UX factors related to the product:** This activity determines which UX factors or quality criteria are to be measured in the further course of development. The method to be developed aims to select from a pool of UX factors only those UX factors that are relevant for the product and the user.
3. **Define a sprint UX goal:** To measure whether a UX goal has been achieved, that goal must be established. In this activity, a UX goal is defined for a period of time. This can include one or more sprints, for example. The purpose is divided into sub goals, which in turn refer to the list of UX factors from Activity 2.
4. **Determine a target UX value for each requirement:** Each requirement is evaluated against the list of UX factors from Activity 2, as it supports the goal. Consequently, it can be decided which requirements should be processed in the next iteration.
5. **Develop a Prototype:** In this activity, the prototype (artefact) is created by the requirement selected in Activity 2. Various methods, such as Design Thinking [16] or Design Sprint [13], can be used. As a result, a low or high fidelity prototype is developed.

6. **Evaluating prototype:** The UX is evaluated based on the UX factors selected in this activity. The result is then compared with the goal defined in Activity 3. From this, conclusions can be drawn for the subsequent development.
7. **Develop the product increment:** In this activity, the product increment is developed. The results from Activities 3 and 6 are considered.
8. **Evaluating product increment:** After the product increment has been developed, the UX is re-evaluated based on the selected UX factors.
9. **Conducting a UX retrospective:** Finally, this activity checks whether the UX goal, as defined in Activity 3, has been achieved. The results of the different evaluations of the UX, based on the selected UX factors, serve as a basis. A comparison is made between the defined UX goal (Activity 3), the evaluated UX of the prototype (Activity 6), and the product increment (Activity 8). This serves as a basis for possible improvements, which can then be applied in the next iteration by, which begins again with Activity 3. If necessary, the next iteration begins with Activity 2 by recreating the list of UX factors.

This process model is to be evaluated as a draft. Further iterations and evaluations are needed to refine the process model. Besides, the process model must be checked to see whether integration within agile software development methodologies is feasible. Also, it should be examined whether a software tool should be developed to support the process model.

6 Conclusion and future work

This paper presents the proposal and the current state of my Ph.D. thesis. The primary objective of the Ph.D. thesis is the development of an iterative and incremental process model. With this process model, development teams should be able to improve product quality concerning UX.

For this purpose, five research questions are asked. RQ1 is answered through an SLR. On this basis, the existing requirements for the process model are further refined and validated (RQ2). Afterward, the process model is developed under consideration of the requirements (RQ3). In a real-world application, the process model must prove itself (RQ4). Feedback then flows into the further development or correction of the process model. In the final step, the process model is evaluated (RQ5).

This Ph.D. thesis contributes to the knowledge base of software development by providing a process model that offers an iterative improvement of product quality in terms of UX and process quality.

References

- [1] David J. Anderson. 2010. *Kanban. Successful evolutionary change for your technology business*. Blue Hole Press, Sequim, Washington.
- [2] Kent Beck and Cynthia Andres. 2007. *Extreme programming explained. Embrace change* (2. ed., 6. printing). The XP series. Addison-Wesley, Boston.

- [3] B. Boehm and R. Turner. 2003. Using risk to balance agile and plan-driven methods. *Computer* 36, 6, 57–66. DOI: <https://doi.org/10.1109/MC.2003.1204376>.
- [4] Manuel Brhel, Hendrik Meth, Alexander Maedche, and Karl Werder. 2015. Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology* 61, 163–181. DOI: <https://doi.org/10.1016/j.infsof.2015.01.004>.
- [5] Tiago S. Da Silva, Milene S. Silveira, Frank Maurer, and Fábio F. Silveira. 2018. The evolution of agile UXD. *Information and Software Technology* 102, 1–5. DOI: <https://doi.org/10.1016/j.infsof.2018.04.008>.
- [6] Marc Hassenzahl. 2001. The Effect of Perceived Hedonic Quality on Product Appealingness. *INTERNATIONAL JOURNAL OF HUMAN-COMPUTER INTERACTION*, 13(4), 481–499.
- [7] Andreas Hinderks, Martin Schrepp, Francisco J. D. Mayo, María J. Escalona, and Jörg Thomaschewski. 2019. Developing a UX KPI based on the User Experience Questionnaire. *Computer Standards & Interfaces*. DOI: <https://doi.org/10.1016/j.csi.2019.01.007>.
- [8] Andreas Hinderks, Martin Schrepp, and Jörg Thomaschewski. 2018. A Benchmark for the Short Version of the User Experience Questionnaire. In *Proceedings of the 14th International Conference on Web Information Systems and Technologies*. SCITEPRESS - Science and Technology Publications, 373–377. DOI: <https://doi.org/10.5220/0007188303730377>.
- [9] ISO9241-210. 2010. *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems*, ISO 9241-210:2010. Retrieved from.
- [10] ISO9241-220. 2016. *Ergonomics of human-system interaction - Part 220. Processes for enabling, executing and assessing human - centred design within organizations*, ISO/DIS 9241-220:2016. Retrieved from.
- [11] Paul Johannesson and Erik Perjons. 2014. *An Introduction to Design Science*. Springer International Publishing, Cham.
- [12] B. Kitchenham and S. Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Retrieved from.
- [13] Jake Knapp, John Zeratsky, and Braden Kowitz. 2016. *Sprint. How to solve big problems and test new ideas in just five days* (First Simon & Schuster paperback export edition). Simon & Schuster, New York.
- [14] Richard Knaster and Dean Leffingwell. 2019. *SAFe distilled. Applying the Scaled Agile Framework for lean enterprises : SAFe 4.5*. Scaled Agile, Inc, [Boulder, Colorado USA].
- [15] Craig Larman and Bas Vodde. 2017. *Large-scale scrum. More with less*. Addison-Wesley, Harlow.
- [16] Tilmann Lindberg, Christoph Meinel, and Ralf Wagner. 2011. Design Thinking: A Fruitful Concept for IT Development? In *Design Thinking*, Christoph Meinel, Larry Leifer and Hasso Plattner, Eds. Understanding Innovation. Springer, Heidelberg, 3–18. DOI: https://doi.org/10.1007/978-3-642-13757-0_1.

- [17] Donald A. Norman. 2007. *Emotional Design. Why We Love (or Hate) Everyday Things*. Basic Books, New York.
- [18] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2014. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24, 3, 45–77. DOI: <https://doi.org/10.2753/MIS0742-1222240302>.
- [19] Jenny Preece, Yvonne Rogers, and Helen Sharp. 2015. *Interaction design. Beyond human-computer interaction* (4. ed.). Wiley, Chichester.
- [20] T. J. Reynolds and J. Gutman. 1988. Laddering theory, method, analysis, and interpretation. *Journal of Advertising Research* 1988, 1, 11–31.
- [21] Dina Salah, Richard F. Paige, and Paul Cairns. 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 - EASE '14*. ACM Press, New York, New York, USA, 1–10. DOI: <https://doi.org/10.1145/2601248.2601276>.
- [22] Eva-Maria Schön, Jorge Sedeño, Manuel Mejías, Jörg Thomaschewski, and María J. Escalona. 2019. A Metamodel for Agile Requirements Engineering. *JCC* 07, 02, 1–22. DOI: <https://doi.org/10.4236/jcc.2019.72001>.
- [23] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. 2014. Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios. In *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Alfred Kobsa, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Demetri Terzopoulos, Doug Tygar, Gerhard Weikum and Aaron Marcus, Eds. Lecture Notes in Computer Science. Springer International Publishing, Cham, 383–392. DOI: https://doi.org/10.1007/978-3-319-07668-3_37.
- [24] Ken Schwaber. 2004. *Agile project management with Scrum*. Microsoft professional. Microsoft Press, Redmond, Wash.
- [25] Pedro Serrador and Jeffrey K. Pinto. 2015. Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management* 33, 5, 1040–1051. DOI: <https://doi.org/10.1016/j.ijproman.2015.01.006>.
- [26] Tiago Silva da Silva, Angela Martin, Frank Maurer, and Milene Silveira. 2011 - 2011. User-Centered Design and Agile Methods: A Systematic Review. In *2011 AGILE Conference*. IEEE, 77–86. DOI: <https://doi.org/10.1109/AGILE.2011.24>.
- [27] Oleg Svatoš and Václav Řepa. 2016. Working with Process Abstraction Levels. In *Perspectives in Business Informatics Research*, Václav Řepa and Tomáš Bruckner, Eds. Lecture Notes in Business Information Processing. Springer International Publishing, Cham, 65–79. DOI: https://doi.org/10.1007/978-3-319-45321-7_5.
- [28] Noam Tractinsky. 1997. Aesthetics and apparent usability. In *the SIGCHI conference*, 115–122. DOI: <https://doi.org/10.1145/258549.258626>.

- [29] Arnold P.O.S. Vermeeren, Effie Lai-Chong Law, Virpi Roto, Marianna Obrist, Jettie Hoonhout, and Kaisa Väänänen-Vainio-Mattila. 2010. User experience evaluation methods: current state and development needs. *Proceedings: NordiCHI*, 521–530.
- [30] Roel Wieringa. 2009. Design science as nested problem solving. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology - DESRIST '09*. ACM Press, New York, New York, USA, 1. DOI: <https://doi.org/10.1145/1555619.1555630>.
- [31] Roel J. Wieringa. 2014. *Design Science Methodology for Information Systems and Software Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg.