

Human–Computer Interaction Series

Editors-in-chief

Desney Tan
Microsoft Research, USA

Jean Vanderdonckt
Université catholique de Louvain, Belgium

More information about this series at <http://www.springer.com/series/6033>

Gilbert Cockton • Marta Lárusdóttir
Peggy Gregory • Åsa Cajander
Editors

Integrating User-Centred Design in Agile Development

Editors

Gilbert Cockton
School of Design
Northumbria University
Newcastle upon Tyne, UK

Peggy Gregory
School of Physical Sciences and Computing
University of Central Lancashire
Preston, UK

Marta Lárusdóttir
School of Computer Science
Reykjavik University
Reykjavik
Iceland

Åsa Cajander
Department of Information Technology
Uppsala University
Uppsala, Sweden

ISSN 1571-5035

Human–Computer Interaction Series

ISBN 978-3-319-32163-9

ISBN 978-3-319-32165-3 (eBook)

DOI 10.1007/978-3-319-32165-3

Library of Congress Control Number: 2016954424

© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature

The registered company is Springer International Publishing AG Switzerland

Foreword

Software thinking is broken and it is not easily fixed. Agile sailed in about a decade and a half ago, but it fell short in many ways. It turned out that the users were completely left out and the customer was brought in as the major source for innovative ideas and creative thoughts about the end-user needs and desires. The software industry was too busy fixing the problems generated by engineering thinking dating back several decades. Yes, we have acknowledged now that big systems cannot be defined in detail upfront and we need to be flexible as we go along. In one way, the agile movement has been a success as, for example, today agile thinking penetrates also to the design of safety critical systems. On the other hand, despite hundreds of studies, we still lack a clear definition of what agile is and what agile is not. Luckily this book does not attempt to answer this definition-related question but addresses something much more fundamental and an important issue, namely, why is it so hard to put together a way of developing software that delivers creative, fun, friendly and easy-to-use software.

This book is rooted in the user-centred design (UCD) field and ultimately wants to combine the great things in agile development with the great things in UCD but does it in an intriguing way that makes the reader wonder: why had I not thought about that before? I have a strong background in software process improvement and in agile software development fields. I have published dozens of industrial studies trying to understand why and how agile works like it does. I thought I could guesstimate upfront the solutions that this book set out to deliver. The authors of this book surprised me very positively. They have opted not to attempt to deliver a single solution that easily fixes all or most of the current problems. Authors share years and years of lessons learned and take a critical standpoint on contemporary thinking. The authors' own field receives a healthy dose of criticism as well as agile development. The book forms a foundational understanding on the complexities surrounding UCD in agile contexts and does it well.

The book has a number of audiences that will benefit for the authors' efforts. I personally will use this book next year when I teach software engineering to 2nd-year undergraduate students of informatics and computer science. The concrete hints, practices and techniques about the user-centred work will benefit students

who think that ultimately Scrum is sufficient for all their needs. Academics more broadly will benefit from the wide angle of perspectives and the critical tone that the book takes when discussing how to make UCD work within agile development. For industrial readers, the book contains a great deal of concrete empirical guidance, as it is based on a large number of industrial case studies in various contexts, companies and cultures.

If I were to interpret the book's solution in making UCD work in agile contexts, it would be its acknowledgement of the fact that no existing methods provide a comprehensive solution in any particular situation, but rather a mixed-method approach is required: companies need to develop their own ways of working, supported by professional training and coaching. The result is a method or a way of working that only works in one organisational setting, since each development situation and the people therein are unique, and that software development is a predominantly intellectual activity where classical engineering methods and tools have proven to only work suboptimally.

Personally, I enjoyed reading especially the forward-looking chapters and tend to agree with the authors that design research and creative thinking would have a lot to give to software developers and researchers. I would have loved to see the authors commenting on the newly formed SEMAT initiative (Software Engineering Method and Theory; see <http://semat.org/>), which is said to form the new theoretical grounding for software development. Although I can understand that SEMAT presents an overly engineering type of thinking, it may not be radical enough to change the status quo in industrial practice.

Is software thinking still broken after having read the book? Perhaps yes, but to a lesser extent. This book shows that there is room for innovative thinking in the field, and I hope readers will agree and find the book as valuable as I have.

Department of Computer and Information Science
Norwegian University of Science and Technology, NTNU
Trondheim
Norway
May 2016

Pekka Abrahamsson

Book Overview

This book originates from a NordiCHI 2014 workshop [1]. Six workshop position papers have been updated and expanded for this book: five case studies (Chaps. 2, 3, 4, 5 and 6) and a proposed new framework (Chap. 9). One position paper [2] was updated for publication elsewhere [3]. The other is available from the workshop's website [4]. Five additional chapters were prepared for this book: the introduction (Chap. 1), the report on the workshop (Chap. 8), a sixth case study (Chap. 7) and two forward-looking analyses (Chaps. 10 and 11). Half of the case studies have industrial authors, with the other half authored by academic researchers working in close collaboration with commercial and public organisations.

The editors' introduction (Chap. 1) begins with the workshop's context; then reviews topics across chapters, position papers and workshop activities; and lastly presents agendas for further research into the effective integration of user-centred design (UCD) and agile development methodologies (Agile).

After briefly overviewing UCD, Agile and their integration (Agile UCD), the introduction surveys four major topics from the workshop and book chapters:

- *Cultures* across the development of digital products and services
- *Teams*: roles, responsibilities, communication, boundaries and capabilities
- *Tasks*: process, sprint and activity-level issues and ideas
- *Research approaches*: human science rigour and creative design research practices

Culture is discussed via values that materialise through people, practices, places and artefacts. *Team factors* are discussed in terms of the following: range of observed roles and associated responsibilities; team boundaries and communication practices; and nurturing cross-functional capabilities. *Task factors* are reviewed across three scopes: processes, iterations and activities. The book's case studies resolve several tensions between Agile and UCD by adapting process and iteration structures, supported by innovative approaches and resources for Agile UCD.

High-quality appropriate research approaches are a major strength of all chapters. Human science traditions within UCD are reflected in rigorous systematic qualitative and quantitative research. Creative design research practices are also in

evidence. These are not all currently common in applied Agile or UCD research but have been applied in several guises to make innovative contributions to Agile UCD practice.

These main topics cover a range of successes and challenges within Agile UCD, as well as future research opportunities. The introduction's first research agenda is practice based and tactical. It can be advanced through professional action research, potentially supported by academic collaboration. The second is more wide ranging, open-ended and foundational. Challenges, issues and questions here reflect the broadening scope of development and operational support for digital products and services, which now are coming close to being the backbone of contemporary commercial and public organisations, rather than simply being internal systems that support work within them. The second research agenda looks beyond Agile and UCD to the wider contexts of digital economies and digitally enabled social and cultural innovation. Chapter 1 recognises the scale of challenges here: collaboration across professions and disciplines is needed to meet them. Fortunately, case study chapters offer good foundations for meeting these future challenges, supported by the forward-looking chapters.

Chapter 2 reports the first of six case studies: *User Integration in Agile Software Development Processes: Practices and Challenges in Small- and Medium-Sized Enterprises* (Oliver Stickel, Corinna Ogonowski, Timo Jakobi, Gunnar Stevens, Volkmar Pipek and Volker Wulf) reports findings on Agile UCD from in-depth case studies within three German SMEs. The research collected data using interviews and observations, which were analysed thematically, drawing on grounded theory. Findings identify three main themes that characterise how SMEs integrate UCD into Agile work:

1. Roles
2. Channels and tools
3. Filtering and interpretation

Chapter 2's recommendations include:

- Understanding the importance of agile and organisational culture
- Holistic consideration of roles, channels and tools
- Awareness of the challenges of filtering and interpreting user feedback

Chapter 2 reveals the diversity of successful Agile UCD, with each SME exploiting a range of roles and practices, especially in relation to customer and user feedback. The mass market orientations of all three SMEs create strong user-focused cultures. All made effective use of shared media and tools, but this was not without challenges.

Chapter 2 balances positive attitudes towards UCD with the recognition that, even in companies with strong user-centred cultures, UCD needs to change. UCD is seen as addressing real users' needs and wants, but not necessarily with clarity on how users should actually be involved and how this fits into established development processes. UCD is not yet mature, because optimal combination of methods is still not well understood. UCD is not the sole source of user-focused practices: good

contributions can be made using management research on innovation, as well as established brand evaluation practices such as Net Promoter Scores.

Chapter 2 identifies where unmodified agile practices such as user stories [5] and daily stand-up meetings support UCD work. Conversely, some innovation practices such as a lead user group have not worked well previously with one SME, and another has concerns about their test households becoming blind to some issues. Overall, this chapter reports sophisticated integrative practices for Agile UCD that allow optimism in the face of more superficial analyses of incompatibility. There are challenges, especially with the product owner role. While the diversity across the three SMEs may obstruct generalisation, Stickel and colleagues rightly argue that the insights from the SMEs' practices can still be helpful.

Chapter 3 presents the second case study: *Templates: A Key to Success when Training Developers to Perform UX Tasks* (Tina Øvad and Lars Bo Larsen) reports the iterative development of curricula and method templates for training developers to carry out UX tasks. Companies often lack staff who are trained in UX and usability methods or do not have enough UX staff to avoid UX teams being perceived as a bottleneck (plus there are also issues about how time consuming UCD work can be). Øvad and Larsen trained developers from three companies with the overall goal of developing a toolkit that was suitable for use within Scrum sprints and would also develop a shared language within development teams. Developers received 1 day of training for each method and were provided with templates that describe how to conduct UX tasks. These templates supported a 'fill in the blanks' approach to UX work. Three methods were covered: focused workshops, AB testing and contextual interviews.

Chapter 3 draws on long-standing UCD research from the 1990s on training developers (especially novices, who need structure) and more recent research within Agile contexts, including work by the authors and collaborators. Agile contexts need to integrate UX work on a daily basis, which was not considered in related 1990s UCD research. A rigorous empirically informed iterative process was followed over 2 years, with up to four iterations for one method. The limitations of 'observe and learn' tactics soon became apparent, so existing template usage, which had been observed in use for documentation in agile development, was transferred to method training materials. Experiences led to continuous changes to templates and guidance documentation.

The effectiveness of the training was rigorously monitored using a mixed methods research approach. Developers were interviewed before and after training and a third time after independent use of the method. Training sessions were observed and recorded for later analysis. Meaning condensation was used for qualitative analysis.

The materials are not intended for stand-alone use but require prior formal training. Introduced in this way, the materials were highly valued by developers, who developed confidence and a secure trust in their capabilities for independent use. However, not all suggestions and requests from trainees were acted on. Iterative improvements were focused on performance before preference. Requests

for examples were not acted on, because of a risk of superficial learning leading to inappropriate copying.

Chapter 4 provides the third case study: *Integrating Scrum and UCD: Insights from Two Case Studies* (Alvaro Aranda Muñoz, Karin Nilsson Helander, Thijmen de Gooijer and Maria Ralph) presents two case studies showing how UCD can be integrated with Scrum. The first case study reports on how a UCD-focused research team proactively and independently adopted and adapted Scrum in two-week sprints to develop a 3D prototype for improving information visualisation for manufacturing.

Team members in the authors' company typically work on more than one research and development (R&D) project. This gave the authors access to other R&D projects for their second case study, which reviewed Agile UCD practices in three other teams. Knowledge, insights and experiences from the two case studies inform the authors' recommendations for future Agile UCD practices:

- Configure physical space to support collaboration, synchronisation and shared understandings across the extended project team, as well as to recognise contributions to ongoing and completed tasks.
- Synchronise resources to support advance planning.
- Plan for UCD's need for extra planning and communication relative to Scrum.
- Plan for external UCD dependencies, especially relating to user research and testing.

Based on the above, the authors propose to modify Sy's approach [6, 7] for Agile UCD, as well as to further adapt Scrum practices:

- Contextual inquiry is restricted to Sprint 1, and completed there, with no requirements for new features collected after that.
- Design chunks (concept groups [6, 7]) can be implemented across a few sprints.
- Anticipating resource needs and aligning these with availability, with some planning several sprints ahead to reduce postponement of UCD tasks.
- Variable sprint lengths.
- Dropping the closed window rule [5] to add tasks to the backlog when developers finish early ('bottomless sprints'), and also when tasks needed to be reprioritised, even within short two-week sprints.
- Use of persistent media to include team members who work part-time on a project, and thus cannot attend all meetings, plus adaptations to Scrum boards to recognise the contributions of all team members (not only developers).

Overall, much of the focus and effort in this case study was directed towards ensuring that all team members were included and valued. Another predominant focus was on being realistic about the uncertainties of creative UCD work in an R&D environment while at the same time anticipating and heading off scheduling problems.

Chapter 5 presents a fourth case study: *Integration of Human-Centred Design and Agile Software Development Practices: Experience Report from an SME* (Carmelo Ardito, Maria Teresa Baldassarre, Danilo Caivano and Rosa Lanzilotti)

presents a planned integration of Scrum and human-centred design (HCD, which unlike UCD, does not see users as design's only stakeholders).

Ardito and colleagues' previous experiences of integrating HCD into a waterfall process created positive expectations, as did principles common to Agile and UCD: iterative design, user involvement, continuous testing and prototyping. While not common to all UCD or Agile, these nevertheless indicate potential for integrating complementary approaches, which was explored collaboratively with experienced project managers from an Italian SME. Thorough literature research on Agile UCD spanning almost a decade of studies identified sources of opportunities and challenges:

1. Upfront tasks, with HCD needing sufficient time here for stakeholder research
2. Prototyping for rapid evaluation and communication
3. User stories, potentially extended to cover usability and acceptance criteria
4. Inspection evaluations of paper prototypes in support of design refinement
5. User testing of interactive prototypes, possibly as part of acceptance testing
6. One sprint ahead for the first few sprints only

HCD could benefit from Agile's high iteration frequency, constant customer involvement and incremental development. Knowledge of opportunities and challenges informed co-design of an agile HCD methodology with points of difference from Scrum:

1. A *customer committee* supports the product owner (PO), increasing customer and user involvement in planning.
2. Multidisciplinary research during project *inception* (UCD, market, technical).
3. An initial *Sprint n.0* resulting in a high-level prototype and basic software services in place, drawing on earlier multidisciplinary research.
4. Physical *Scrum Islands* sitting two developers, visual interaction designer and PO together, removing need for daily stand-up meetings and formal sprint end reviews.
5. Predefined very short one-week sprint cycles, with tasks carrying over to next sprint.
6. Continuous *(IN)Sprint Reviews* supported by customer committee testing.
7. *Multidisciplinary project retrospective* after project completion, covering product quality and customer satisfaction and opportunities for the SME to improve strategy, management and software and process quality.

Customer feedback was gathered throughout the project from the earliest stages, with customers actively engaged at multiple points throughout, including acceptance testing for both usability and internal software quality. The extensions here were experienced as a positive integration of HCD into Scrum-based practices.

Chapter 6, *Communication Breakdowns in the Integration of User-Centred Design and Agile Development* (Silvia Bordin and Antonella De Angeli), reports the fifth case study of the large university-based Smart Campus project. This was moved to Scrum from an initial iterative participatory design (PD) process. PD is a form of UCD where 'the people destined to use the system play a critical role

in designing it' [8]. This happened almost 1 year into the project, forcing dynamic adaptation of Agile and PD practices, adding to the challenges for Agile UCD. The corresponding position paper's title was *Catch Me If You Can: Reconciling Agile and UCD*, which refers to the challenge of reconciling PD with Agile at a fast pace in the context of a high volume of feedback from a large user community.

Smart Campus aimed to develop mobile apps to support a student community. A pervasive PD approach was expected, supported by multiple feedback channels. Before Scrum was introduced, extensive upfront work had preceded app development. However, the unanticipated introduction of Scrum made it difficult to fully follow PD principles. To understand the difficulties encountered, and how to overcome them, two interview studies were planned and carried out. A researcher external to the project conducted the interviews, which were then transcribed and analysed, controlling for coding bias. A literature review in conjunction with the interview studies revealed how UCD work practices can be obstructed by Agile; the authors benefitted from two 2014 literature surveys that covered 76 [9] and 71 [10] papers, which highlight the topicality of the chapters in this book.

Three themes emerged from the combined literature and interview data analyses:

1. Differences in how user involvement is understood across roles in the project team
2. Differences in how documentation was valued by project team roles
3. Coordination of design and development work

The nature and funding of Smart Campus resulted in management approaches different to those advocated by Scrum: a wider range of stakeholders were actively involved and forced breaches of the Scrum principle of self-organising teams [5]. Similar breaches are reported in other case studies, without evidence of any negative impact. However, PD involving a large user community significantly adds to the challenges for UCD within Scrum. Bordin and De Angeli thus advocate adopting participatory development [8] and design thinking [11, 12] alongside Scrum to promote an organisational culture receptive to innovative software design for communities.

Chapter 7 presents the last study: *Towards Understanding How Agile Teams Predict User Experience* (Kati Kuusinen, Heli Väättäjä, Tommi Mikkonen and Kaisa Väänänen). It is an inventive study of how accurately different roles with an agile team can predict a system's UX from users' perspectives. As with the training courses and materials developed by Øvad and Larsen (Chap. 3), the motivating practical goal is to reduce UX specialists (UXSs) being (perceived as) a bottleneck. If developers can perform more UCD work, then faster design iterations based on UX evaluation are possible, and meaningful UX goals can be set for sprints. For example, another study for the projects involved [13] indicated that developers did not participate in user tests or identifying and defining target user groups.

Team members from six enterprise software development projects in five companies participated. All six used Agile with release cycles under 6 months long and released software in use. The application contexts were work-based enterprise

systems. Software had graphical user interfaces, requiring UX design work that was underway.

Each participant rated their enterprise application against 16 UX dimensions and also gave it an overall UX rating and rated how well it responded to needs. Need fulfilment ratings by team members correlated most strongly with users' ratings for non-instrumental UX, e.g. 'aesthetic' or 'presentable'. Only users' ratings for 'useful' correlated significantly with their assessment of need fulfilment. Team members rated their released software twice: once from their own perspective and once from the users. Ideally, these ratings should match, but the former was more critical than the latter. However, PO and UXS ratings from the users' perspective were closer than developers' to users' actual ratings.

The results show that team members can predict instrumental aspects of UX but are less able to predict hedonic quality, with POs and UXS performing better than developers. To better predict UX ratings, developers need to better understand users. Chapter 7 identifies the use of personas as one possible tactic here. Without better knowledge of users, making use of developer ratings of achieved UX could be harmful.

The remaining four chapters are forward looking and authored by academic researchers but all draw on industrial case studies. Chapter 8 uses the combined input of workshop attendees. Chapters 9 and 10 use retrospective analyses of several case studies. Chapter 11 uses secondary sources from research into creative design.

Chapter 8 reports on the NordiCHI 2014 workshop from which this book originates. It describes its motivation, the approach used, its eight position papers and analysis and discussion of its themes. Six position papers were Scrum case studies. The others focused on Kanban [4] and Scrum and Lean [2].

Agile-style techniques were used to run the workshop, including time-boxed increments to workshop analyses, a small group discussion approach and a persistent shared visual workspace. During the workshop, post-it notes were collected from all participants about 'challenges and obstacles' and 'interesting points' that they had identified during position paper presentations. Post-its were then organised into two affinity diagrams. The smallest affinity groups for 'challenges and obstacles' were 'tools/toolboxes' and 'synchronisation', indicating that tools and high-level process issues were modest concerns (even though synchronisation issues dominate the Agile UCD literature). The smallest post-it groups for 'interesting points' were 'tools' (again), 'documentation' and 'UX team' (does that reflect attendees' UX credentials?). Large groups for 'interesting issues' formed around user involvement and feedback, pair working and internal communication. Chapter 8 presents results from a phased affinity diagram analysis of the post-its. Eight final themes were formed:

- People and roles
- Teams and communication
- Culture ('challenges and obstacles' only)
- Methods and practices
- Time and synchronisation

- Artefacts and tools
- Research and problems ('interesting points' only)
- Miscellaneous

The largest theme by post-it count, 'Methods and practices', highlighted the practicalities of integrating UCD into Agile and the practices that make that possible. Two people-centred themes, 'Teams and communication' and 'People and roles', came next. Both are key to improving integration. The smallest theme, 'Artefacts and tools', was nevertheless the focus of some novel solutions presented at the workshop. These themes reflect the scope and breadth of discussions, with much focus on fitting together the big picture of agile theory and methods with the lower-level detail of day-to-day practices. Potential links between theory and practice provided a focus for considering innovative new practices to ameliorate persistent challenges to effective Agile UCD. However, despite the workshop's focus on research and innovation as well as challenges, discussion focused on familiar unresolved issues, despite many presented examples of progress. Nevertheless, some new challenges and innovative solutions do not figure prominently in current surveys:

- Power relations in Agile
- Training developers in UCD ('Developers doing UCD' was the largest initial 'interesting issues' group)
- Filtering and interpretation of user feedback

Chapter 9 presents Kati Kuusinen's *BoB: A Framework for Organizing Within-Iteration UX Work in Agile Development*. This new framework seeks to combine the 'best of both' worlds of UCD and Agile. BoB uses an initial upfront activity for early product definition, which includes a workshop and results in a clickable version of an initial product. After this important enabling milestone (similar to Chap. 4's Sprint 1 and Chap. 5's Sprint n.0), a sequence of sprints requires a cross-functional team to work together, thus avoiding difficulties with the one sprint ahead approach [6, 7], which tend to result in inescapable mini-waterfalls across sprints. The BoB framework is based on four mixed methods studies spanning 4 years and involving over 300 respondents from 9 companies in 10 countries (7 in Europe and 3 in Asia), working across IT services, engineering, middleware, mobile enterprise applications and industrial systems (including safety critical). The BoB framework is supported by guidelines on people, process, tasks and tools. It shifts the focus from roles (as in Scrum) to analysis and design tasks, which are carried out by small cross-functional teams. It reduces the emphasis on 'definitions of done' (as in Scrum [5]) with an acceptance of trial and error, with the need to iterate user interface designs [14].

Elements of the BoB framework are already in use by some companies involved in the underpinning studies. It is too early to judge whether BoB will overcome challenges for Agile UCD, but Kuusinen expects cross-functional teams to work together better and that faster feedback from customers and users will result. BoB is expected to halve the feedback cycle time for a feature compared to one sprint ahead

Agile UCD, since design and development are concurrent instead of sequential. Thus, it should take one iteration instead of two to receive and act on feedback based on real use.

Chapter 10 provides a refreshing change from UCD criticisms of Agile's shortcomings. *Challenges from Integrating Usability Activities in Scrum: Why Is Scrum So Fashionable?* (Marta Lárusdóttir, Åsa Cajander, Guðbjörg Erlingsdóttir, Thomas Lind and Jan Gulliksen) instead investigates the positive reasons why organisations choose Scrum. The attractiveness of Agile is contrasted with that of UCD, which is not as fashionable. As with Chap. 9 by Kuusinen, this chapter is based on retrospective analysis of existing case studies, in this case data from five survey and interview studies involving 110 respondents from over 40 companies in mainly two countries. The results of these studies are interpreted through the twin lenses of Abrahamsson's theory on management fashion [15] and Roger's diffusion of innovations theory [16]. For much of UCD's history, it has held the moral high ground [17], insisting that we must put users' needs and comfort first. The respondents in the five studies did not all always do so. While there is no doubt that Scrum benefits from its fashionability (much of this of its own making [5]), Abrahamsson's theory on management fashion [15] requires rational and progressive motives for adopting new innovations, and the respondent's positions in the case studies make it clear that, to them, Scrum adoption is both rational and progressive. Scrum also has advantages in relation to Roger's diffusion of innovations theory [16]. The future fortunes of UCD in most software development contexts thus depend on understandings of management fashions and diffusion of innovations. UCD's moral high ground has clearly not won the majority of hearts and minds since the 1980s, so alternative approaches to dissemination and uptake are needed that focus on UCD's worth as a favourable balance of benefits over costs and risks. Both UCD and Agile have understandably focused on promoting their benefits while downplaying or ignoring their costs and risks. Balanced approaches are required.

Chapter 11 introduces a third element to the methodology mix. *Integrating Both User-Centred Design and Creative Practices into Agile Development* (Gilbert Cockton) argues for the integration of creative design practices as well as UCD within agile methodologies. The expectation is that the benefits arising from a more balanced and integrated design process will increase by knowingly adding key creative design practices (rather than assuming that Agile UCD is creative enough). Also, costs and risks will decrease. Cockton draws on almost half a century of secondary literature from design research to identify three key insights on the nature of creative design work:

- Creative design work co-evolves problem and solution spaces.
- Design materials talk back.
- The best design work is generous in scope and intent.

The first insight guides Agile and UCD to fully break away from the constraints of idealised rational engineering design and its rigid segregation of problem and

solution spaces via a thick wall of requirements specifications [18]. The second insight defends the use of UCD and creative design ‘documents’ (as in Chap. 6), which are to be understood in the broadest sense of persistent media rather than Agile’s preferred face-to-face conversations. Such documents are expected to change and this is not wasteful. Instead, this is one of the primary ways through which problem and solution spaces co-evolve in creative design work. The third insight challenges UCD and Agile practices to look beyond the ‘requirements of others’ and let designers add their subjective generosity to the objective needs and wants of users (UCD) or a product owner’s user stories and associated business value (Agile). Acting on these three insights will provide opportunities to balance and integrate creative, engineering and user-centred design and, in doing so, be guided and shaped by designers’ generosity. The result here is a design process that is BIG [19]: balanced, integrated and generous.

Overall, this book’s chapters present a comprehensive survey of progress and continuing challenges in the integration of Agile and UCD. The quality and depth of the case study research is impressive. The forward-looking chapters propose novel broader futures for Agile UCD. There are good grounds for optimism now in the face of the initial shock of Agile reversing many UCD gains. These reversals are temporary, and we can look forward to the best of both worlds, or perhaps the best of several worlds, combining to improve the development of digital products and services.

Newcastle upon Tyne, UK
Reykjavik, Iceland
Preston, UK
Uppsala, Sweden
May 2016

Gilbert Cockton
Marta Lárusdóttir
Peggy Gregory
Åsa Cajander

References

1. Lárusdóttir M, Cajander Å, Gulliksen J, Cockton G, Gregory P, Salah D. On the integration of user centred design in agile development. In: Proceedings of NordiCHI’14. ACM, pp 817–82
2. Law ELC, Lárusdóttir MK (2014) User Experience (UX) design: agile or lean?. In: Proceedings of the workshop: on the integration of UCD and agile development, NordiCHI 2014, Helsinki, Finland. Available from <https://ucdandagile.wordpress.com>
3. Law ELC, Lárusdóttir MK (2015) Whose experience do we care about? Analysis of the fitness of Scrum and Kanban to user experience. *Int J HCI* 31(9):584–602
4. Lindell R (2014) Attending experiential qualities in system development. In: Proceedings of the workshop: on the integration of UCD and agile development. NordiCHI 2014, Helsinki, Finland. Available from <https://ucdandagile.wordpress.com>
5. Meyer B (2014) Agile! The good, the hype and the ugly. Springer
6. Sy D (2007) Adapting usability investigations for agile user-centered design. *J Usability Stud* 2(3):112–132
7. Sy D, Miller L (2008) Optimizing agile user-centred design. In: CHI’08 extended abstracts on human factors in computing systems, CHI EA’08, ACM, pp 3897–3900

8. Schuler D, Namioka A (eds) (1993) *Participatory design: principles and practices*. Lawrence Erlbaum, Hillsdale
9. Jurca G, Hellmann TD, Maurer F (2014) Integrating agile and user-centered design: a systematic mapping and review of evaluation and validation studies of agile-UX. In: *Agile conference (AGILE)*, 2014, IEEE (2014), pp 24–32
10. Salah D, Paige RF, Cairns P (2014) A systematic literature review for agile development processes and user centred design integration. In: *Proceedings of 18th International conference on evaluation and assessment in software engineering*, 5. ACM
11. Brown T (2005) *Strategy by design*, fast company, June 2005. <http://www.fastcompany.com/52795/strategy-design>
12. Brown T (2009) *Change by design: how design thinking transforms organizations and inspires innovation*. Harper Business, New York
13. Kuusinen K (2015) Task allocation between UX specialists and developers in agile software development projects. In: *Proceedings of INTERACT 2015*, LNCS 9298. Springer, pp 27–44
14. Gould J, Lewis C (1985) *Designing for usability: key principles and what designers think*. CACM 28(3):300–311
15. Abrahamsson E (1996) Management fashion. *Acad Manag Rev* 21(1):254–285
16. Rogers EM (2003) *Diffusion of innovations*, 5th edn. Free Press, New York
17. Cooper G, Bowers J (1995) Representing the user: notes on the disciplinary rhetoric of human-computer interaction. In: Thomas PJ (ed) *The social and interactional dimensions of human-computer interfaces*. Cambridge University Press, Cambridge, pp 48–66
18. Gram C, Cockton G (1996) *Design principles for interactive software*. Chapman and Hall, London
19. Cockton G (2013) Design isn't a shape and it hasn't got a centre: thinking BIG about post-centric interaction design. In: *Proceedings of MIDI'13*. ACM, Article 2, 16 pages. doi:[10.1145/2500342.2500344](https://doi.org/10.1145/2500342.2500344)

Acknowledgements

We would like to thank the NordiCHI 2014 workshop chairs, our workshop participants, the authors of chapters, our reviewers and colleagues at Springer for making this book possible. The workshop merged two separate proposals, improving the balance of expertise across the organisers. The COST TwinTide Action IC0904 (www.twintide.org/) provided a context for collaborations between three of the editors.

Contents

1	Integrating User-Centred Design in Agile Development	1
	Gilbert Cockton, Marta Lárusdóttir, Peggy Gregory, and Åsa Cajander	
Part I Case Studies		
2	User Integration in Agile Software Development Processes: Practices and Challenges in Small and Medium Sized Enterprises	49
	Oliver Stickel, Corinna Ogonowski, Timo Jakobi, Gunnar Stevens, Volkmar Pipek, and Volker Wulf	
3	Templates: A Key to Success When Training Developers to Perform UX Tasks	77
	Tina Øvad and Lars Bo Larsen	
4	Integrating Scrum and UCD: Insights from Two Case Studies	97
	Alvaro Aranda Muñoz, Karin Nilsson Helander, Thijmen de Gooijer, and Maria Ralph	
5	Integration of Human-Centred Design and Agile Software Development Practices: Experience Report from a SME	117
	Carmelo Ardito, Maria Teresa Baldassarre, Danilo Caivano, and Rosa Lanzilotti	
6	Communication Breakdowns in the Integration of User-Centred Design and Agile Development	137
	Silvia Bordin and Antonella De Angeli	
7	Towards Understanding How Agile Teams Predict User Experience	163
	Kati Kuusinen, Heli Väättäjä, Tommi Mikkonen, and Kaisa Väänänen	

Part II Future Directions

8	Workshop on the Integration of User-Centred Design and Agile Development: Approach, Findings and Themes	193
	Peggy Gregory, Marta Lárusdóttir, Åsa Cajander, and Gilbert Cockton	
9	BoB: A Framework for Organizing Within-Iteration UX Work in Agile Development	205
	Kati Kuusinen	
10	Challenges from Integrating Usability Activities in Scrum: Why Is Scrum so Fashionable?	225
	Marta Lárusdóttir, Åsa Cajander, Guðbjörg Erlingsdóttir, Thomas Lind, and Jan Gulliksen	
11	Integrating Both User-Centered Design and Creative Practices into Agile Development	249
	Gilbert Cockton	

Contributors

Carmelo Ardito Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Bari, Italy

Maria Teresa Baldassarre Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Bari, Italy

Silvia Bordin Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Danilo Caivano Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Bari, Italy

Åsa Cajander Department of Information Technology, Uppsala University, Uppsala, Sweden

Gilbert Cockton School of Design, Northumbria University, Newcastle upon Tyne, UK

Antonella De Angeli Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Thijmen de Gooijer ABB Corporate Research, Västerås, Sweden

Gudbjörg Erlingsdottir Lund University, Lund, Sweden

Peggy Gregory School of Physical Sciences and Computing, University of Central Lancashire, Preston, UK

Jan Gulliksen KTH Royal Institute of Technology, Stockholm, Sweden

Karin Nilsson Helander ABB Corporate Research, Västerås, Sweden

Timo Jakobi University of Siegen, Siegen, Germany

Kati Kuusinen Tampere University of Technology, Tampere, Finland

Rosa Lanzilotti Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Bari, Italy

Lars Bo Larsen Aalborg University, Aalborg, Denmark

Marta Lárusdóttir School of Computer Science, Reykjavik University, Reykjavik, Iceland

Thomas Lind Department of Information Technology, Uppsala University, Uppsala, Sweden

Tommi Mikkonen Tampere University of Technology, Tampere, Finland

Alvaro Aranda Muñoz ABB Corporate Research, Västerås, Sweden

Corinna Ogonowski University of Siegen, Siegen, Germany

Tina Øvad Radiometer Medical, Copenhagen, Denmark

Aalborg University, Aalborg, Denmark

Volkmar Pipek University of Siegen, Siegen, Germany

Maria Ralph ABB Corporate Research, Västerås, Sweden

Gunnar Stevens Bonn Rhein-Sieg University of Applied Sciences, Augustin, Germany

Oliver Stickel University of Siegen, Siegen, Germany

Kaisa Väänänen Tampere University of Technology, Tampere, Finland

Heli Väättäjä Tampere University of Technology, Tampere, Finland

Volker Wulf University of Siegen, Siegen, Germany