Customer Feedback and UCD in Agile Software Development

Oliver Stickel, Sebastian Draxler, Gunnar Stevens
University of Siegen
Hoelderlinstr. 3, 57076 Siegen, Germany
oliver.stickel / sebastian.draxler / gunnar.stevens
@uni-siegen.de

ABSTRACT

In this contribution, we address issues relating to the area of tension between User Centered Design (UCD) and agile software development. It reflects on a case study in a German software development company that focused on how this company integrates customer feedback and UCD into their agile practices. We identified aspects crucial for UCD as involved roles, channels & media for customer input as well as the filtering and selection mechanisms regarding this input. Against the background of the study as well as existing work, we then propose some issues, questions and input for discussions for the workshop on the integration of UCD and agile development at NordiCHI 2014.

Author Keywords

User Integration, Agile, Development Process, Scrum, Usability, User Experience, Field Study, Qualitative Research, User Centered Design

ACM Classification Keywords

D.2.9 Software Engineering: Management

INTRODUCTION

Integration of the customer's wishes, needs, input and feedback into software development on a continuous basis is crucial in order to work towards maximizing the usability and uxer experience (UX) of a software product as emphasized in ISO 9241-210. Software development itself becomes increasingly more agile which results in the necessity to integrate agile development methods and UCD in order to achieve an integrated perspective on design and use [12]. We are currently working on a research project aimed at helping Small and Medium sized Enterprises (SMEs) to facilitate this integration by ways of supportive ICT, best practices and process model enhancements. To ground our work, we are conducting empirical research in German SMEs in order to understand practices and challenges relating to customer integration, UCD and agile software development *in situ*. At this point, we have completed our first in-depth case study into an evolved, successful agile software development project ("FinanceX", pseudonymized) of a company at the larger end of the SME spectrum ("Mavolio corp.", pseudonymized) with a focus on UCD.

So far, our results indicate three important aspects related to customer integration, UCD and agile development: *Roles* which concern ones found in traditional agile doctrine but also have been found to be expanded into other areas in practice. *Channels and Media*, i.e. conduits and means for communication with the customer but also internally and the *filtering* of customer input (i.e. appraising it, analyzing and understanding it, matching it to internal qualitative or technical goals and similar challenges). We will structure this position paper around this case study as well as the related literature. We will first give a brief overview about the state of the art, introduce our research methods and stance and subsequently report on the results before discussing them and elaborating further on issues, challenges and discussion points relating to UCD and agile development for the NordiCHI 2014 workshop.

STATE OF THE ART

Agile software development [2] refers to relatively new paradigms (e.g. Scrum [13] and Kanban [1]) to structure ICT projects which differ from classical ones (e.g. the waterfall) models in that they reject the notion of a "heavy", largely predefined and pre-planned project layout which is then processed step by step. Instead, Agile methods state that during the course of a software project, there will almost always be change and propose to prepare for this and embrace it [21]. This results in four central values as codified in the *Agile Manifesto* [2]: 1. Individuals and interactions over processes and tools 2. Working software over comprehensive documentation 3. Customer collaboration over contract negotiation 4. Responding to change over following a plan.

The agile methodology utilized by the FinanceX team is *Scrum*. Scrum can be characterized through "Sprints" which are relatively short (usually less than four weeks) development cycles, during which specific features are implemented

and after which a usable version should be completed. Crucial for Scrum is the rule of 3-3-3-3 which stands for roles internal, roles external, artifacts and ceremonies (more in [18]).

The current state of user centered design (UCD) as codified in ISO 9241-210 (more e.g. in [10]) can be seen as a normative design and development model that argues for user/customer integration in several phases of the process. The UCD ideology specifically views the user as an asset of the product development process. Furthermore, it explicitely focuses on generating a positive user experience. It suggests to include users in early idea finding phases, to carry out user research (this covers everything that helps to understand who the users are, what their system of values and requirements are, etc.), (optionally include users in mock-up generation or evaluation) and include users in the evaluation of releases. User centered design was developed to be open enough to be adapted to existing software engineering approaches, as the specification leaves room that has to be filled with interactions between a design team and an engineering team.

It has to be noted that agile methods already seem to incorporate some aspects and resemblances to UCD, e.g. the iterative approaches and the emphasis on the customer [4]. This is also stated in the Agile Manifesto itself, see [2]. There are multiple reports on this relation, e.g. a single case study on a multinational corporation and its shift to agile [7] or [19], a case report which notes that agility in the development process together with adjustments in frequency and reporting of usability testing activities to match the agile cycles has proven beneficial for the usability and UX of their products.

However, it has also been noted frequently (e.g. [8, 6, 9]) that the actual integration of both worlds often does not work as well as it might and is not yet well understood in practice. Hence, there have been multiple scientific workshops and tutorials, e.g. [16] and suggestions for procedural models or frameworks to facilitate the integration, e.g. [14] who base their framework on an Interaction Design Lifecycle and specifically input design cycles into the agile process or [3] who focuses on usability professionals and how to integrate them in agile environments, not least by facilitating understanding for development practices on the UX side of the team. Scrum roles are also regarded as relevant for the successful integration of UCD and agile: Singh [15] identifies the Project Owner (PO) as the most crucial role for such attempts and states that POs often are overwhelmed since they have to coordinate so many stakeholders, artifacts and ceremonies and are frequently not qualified in-depth for usability and UX which leads to the proposal to appoint two POs, one of which focuses on a POs more traditional role while the other ones responsibilities lean towards usability.

METHODOLOGY

Our basic research for this case study was: *How does Mavolio integrate customer input and feedback into their agile software development process and how does this relate to UCD?* It is important to note that we did not approach the field with predefined categories of existing development processes and tools but rather based our modus operandi on Grounded Theory [17]. Therefore, we took our research question and quickly went into the field where we iteratively developed our understanding and research strategy according to our findings. The efforts described in this paper can hence be viewed as stepping stones towards a more comprehensive theory of UCD in agile software development processes. We view such a stance which is driven by the data and the field to be important given the disparities between theory and practice and the ambiguities as described in the state of the art.

No.	Role
I01	Product Owner
I02	Social Media Officer
I03	Chief of Support
I04	Customer Lab (2 interviewees)
I05	First Level Support employee
I06	Software Developer
I07	Product Owner, different software project (contrast)
O01	Observation Scrum sprint planning meeting
O02	Observation of two usability tests

Table 1. Interviews and Observations

For our research, we selected FinanceX, one of multiple software projects in development by Mavolio. FinanceX, a personal finance management system, was chosen because of its market success, positive customer reviews and its internal focus on agility as well as user centeredness, usability and UX. FinanceX also caters to a very broad audience, it is multiplatform and hence was deemed quite challenging regarding appropriate customer integration and UCD requirements. Our reasoning behind choosing such a project was that we expected to gather broader data, identify more prominent challenges or breaking points and practices than in a smaller, less evolved or narrower project.

Three researchers spent a total of about 15 hours *in situ*, where we primarily conducted the following research activities: Seven face to face interviews of between one and two hours each with key participants in Mavolio's development process (see tab. 1). The Project Owner was selected as the first interviewee through its managing role for the software project and the interview with him followed only very basic guidelines. From here on out, the selection of the next interviewees as well as the interview guidelines evolved from the field. Specific queries to past interviewees if new questions or talking points arose from the field. Those were done mostly face-to-face but some via e-mail.We also conducted observations of two sessions of user tests in Mavolio's in-house usability testing laboratory ("CustomerLab") as well as a participant observation of a Scrum Sprint planning meeting.

From those activities, we gathered about twelve hours of audio recordings which we transcribed in full, albeit content-focused, only including breaks, exclamations and similar meta-information if the corresponding occurrence was very prominent. We also collected about 12 pages of handwritten field notes and memos as well as multiple textual and visual artifacts, e.g. usability testing reports from the CustomerLab and screen shots from supportive ICT-tools utilized by Mavo-

lio. All data and artifacts were subsequently coded axially and selectively [17]. The coding process started immediately after the first interview and was continued and evolved all throughout the research activities. Once to twice weekly, we held discussion and mirroring efforts of the coding activities in our research group. This also included four researchers which were not active in the field and helped asking questions towards the data. We also had similar sessions with colleagues who were not involved in our research project at all and provided us with valuable unbiased insights as well as forced us as immersed researchers to explicate a significant amount of tacit information. This permanent condensation of the coding structure was followed up to the point where the gathered data did not add significant new insight (saturation). During this process, several categories emerged from the data on which we will report next.

FINDINGS

In this section, we will report on the three most prominent codes: Roles, Channels & Media and Filtering as well as their interrelations.

Roles

As in established Scrum doctrine, we found the role of the **Project Owner** (PO) to be the central hub of the customer integration and the focus on UCD. However, in the case of FinanceX and deviating from classical Scrum, we found a team of three POs who share their responsibilities: PO1 is the visionary, specifying the *"epics"* (I01), PO2 has more of an executive day-to-day role and PO3 leans towards interaction design. However, all three POs have an eye on the user stories and discuss them collaboratively, sometimes including other members of the development team (e.g. for technical expertise). In order for the POs to do their job, they obviously need to receive customer input. For FinanceX, three roles or departments are primarily responsible for gathering and providing this input:

The **Support Team** (ST), a unified in-house helpdesk. The ST is organized hierarchically with support employees, team leaders and one overall chief of support. The ST is in contact with customers on a daily basis. Over the course of time, there seem to have emerged experts for specific products or product aspects within the ST.

The **Social Media Officer** (SMO) who monitors and engages in communication relevant to the company on Social Media.Mavolio has separate Social Media presences for each of its products as well as one for the company itself.

The **Customer Lab** (CL): An in-house usability lab which handles user tests for all of Mavolio's products on request of the development teams (usually their heads, in FinanceX's case the POs). Tests occur in every stage of the development and with different focusses (e.g. on the UI, on specific features, etc.). Staff members of the CL also take part in Scrum meetings if the CL is involved at that stage of the project.

Channels and Media

The **Product Owners** as the central coordinator utilizes a Microsoft Team Foundation Server (TFS) to manage the FinanceX project. Entries in the TFS are categorized as Bugs, Improvement Proposals (self-implemented, smaller quick-fix alterations) or User Stories and can be prioritized. The TFS is the PO's central channel to the developer team. Other channels include E-Mail, phone and given the moderate size of Mavolio, a lot of personal contact and office grapevine (expressly mentioned in almost all interviews). Externally, the PO team is not in contact with customers on a daily basis, but there is a dedicated public E-Mail address for FinanceX which is read by the POs. Notably, there are a few long-time or very dedicated customers who know the PO's call-through or their company E-Mail addresses and contact them directly if they have suggestions or issues with the product. The POs also monitor the ratings in the different app stores where FinanceX is available.

The **Support Team** offers E-Mail, web-based, chat and phone support to the customers. Furthermore, there are bulletin boards for every product but they are considered as "users helping users" and only loosely monitored by the ST. The most important channels by far are considered to be E-Mail and phone. During support calls or other forms of communication, customers will often make suggestions, report bugs or give other feedback. Mavolio's support employees are also trained to expressly ask for such information. All customer feedback is recorded in a support database where related issues are cross-referenced and counted. If issues are deemed worthy (see "Filtering" for more detail on this process), they can be fed into the TFS. The ST can also in some instances initiate developer activities directly without the detour through the PO, however, this happens only in very specific routine instances (e.g. if a customer suggests a new bank should be integrated into the FinanceX app, which is usually a straightforward and well-established process at Mavolio and does not need to be wrapped into a User Story).

The **Social Media Officer's** most important channels / media are obviously Facebook and Twitter as well as blogs. She constantly monitors all platforms, keeping an overview about what is happening, reading discussions and gathering feedback, error reports and suggestions. If discussions run towards unwanted or wrong angles, the SMO also tries to intervene. In case of problems like server outages or similar issues, the SMO informs the customers over app available channels and, more importantly, keeps them up to date. Internally, the SMO is in close contact with the Support Team via phone and E-Mail and sometimes refers customers to the ST. Similarly to the ST, in some (pre-)defined routine aspects, the SMO also talks directly to the development team and assigns them tasks. However, her central contact are the POs to whom she relays customer feedback and suggestions, utilizing E-Mail as the tool of choice, often supplemented through face to face discussions. Notably, she does not input information into the TFS directly.

The **Customer Lab** primarily utilizes single-user sessions with 5-20 sessions per test battery for a specific question or feature, mostly relying on testing the software while Think-

ing Aloud [5] and being recorded with a camera as well as a screen capturing tool. Sometimes, heuristic evaluations [11] and cognitive walkthroughs [20] are utilized and currently, the CL is branching out towards testing and understanding users in situ at home or in their working environment. After each session, smaller reports are created and fed into the TFS for the POs. After each test battery, a more comprehensive PDF report is created. Notably, it is possible for everybody in the development team to tune into the live feed from the usability testing sessions, although it has been expressed in multiple interviews that developers usually do not do this. Tests in the CL are only carried out on request of the POs, the management or other decision making roles.

Filtering

In the previous sections, we tried to clear up how and with the participation of which roles customer feedback is gathered and passed along through the company. There is, however, one step missing which is what we call *filtering*. This relates to the process of analyzing and assessing user feedback as well as matching it to other feedback or internal goals for the product. It also encompasses on the challenge of identifying what the user *really* means or needs. These aspects have quickly emerged in our interviews as so important that they deserve deeper analysis.

Mavolio has a long history of experimenting with the incorporation of user feedback and UCD in their development cycles and within this history, there have been failures, too. One example from the interviews (I01, I03, I05) is a former project where the development of a software product relied very heavily on the input of a selected group of users which were considered lead users for the area. However, as it turned out later, the product became much too specialized and thus didn't appeal to many potential customers. Experiences like this reinforced Mavolio's focus on filtering as well as spreading out the sources for the input – as outlined in the previous sections, customer feedback is sampled through a wide range of channels, representing an attempt to level the playing field and keep specialization appropriate to the product.

The most important decision makers regarding the filtering process are obviously the **POs**. They try to match their vision of the product with the customer input, adapt, prioritize and, if deemed necessary, modify or deny specific feedback. The **CL** does not filter actively at all, its staff simply aggregates their results and hand them to the POs. The **SMO** also does not really filter actively – she does however match every incoming input with previous decisions made by the POs and if the input is identical or very similar, she informs the customer accordingly (e.g. request denied, request in development, etc.) without alerting the POs. If a specific feedback is new and she hands it over to the POs, she usually annotates it and states her opinion about it, i.e. gives her input into the filtering process. Most notably is the **ST** which *does* filter actively. Customer feedback is gathered and discussed between the leader of the support team and the chief of support and filtered. Thus, some feedback may never even reach the POs.

Consistently through all interviews, the exact operationalization of the filtering process remains somewhat indistinct but can be categorized broadly into two classes: *Qualitative* and *Quantitative* Filtering. Quantitative filtering concerns the frequency and intensity of a specific type of feedback. Especially the ST seems to utilize quantitative filtering which is also easy to do for them since the customer feedback from each call[1] is recorded in their database. Quantitative aspects are, however, no guarantee for the feedback to get implemented – an often cited example (I01, I02, I03, I04, I05) from our interviews is that of a feature requested by a very large amount of customers but which would make other features planned by PO1 for the future of the product technologically impossible and hence does not get implemented. The other way around, i.e. individual or occasional cases seem more straightforward: (Very) specific features which get requested by very few people usually get filtered out.

Qualitative filtering is a "softer" aspect and seems primarily associated with experience and routine rather than hard data. It was hard for all interviewees to explicate techniques and methods for qualitative filtering. Instead, in nearly every interview, it was stated ex- or implicitly that a lot of know-how with and a "*feeling for*" the product has to be developed over time in order to get it right. A necessity to be a user of the product oneself has also been mentioned. It seems as if the SMO leans more towards qualitative filtering than the other involved roles.

Qualitative and quantitative filtering mechanisms are reported to compliment each other well and none of them is seen as sufficient for itself. Especially the POs seem to try and consider and value both aspects.

DISCUSSION AND OUTLOOK

The involved **roles** seem exhaustive and complementing each other. The trinity of POs in FinanceX's case incidentally mirrors impulses found in the literature as mentioned in the state of the art [15] and it seems that such a division of labor has merit, especially if a software project gets bigger but also calls for more coordination and articulation work through the whole team (e.g. through a Scrum Master). Regarding possible conflicts between internal product vision of the "own" project vs. customer feedback, multiple empowered roles and slightly different perspectives can help to avoid Group Think – evidenced in our case study by heated discussions between PO and SMO. The latter also does not have to focus on administrative aspects which also lead to valuable differences regarding the view of the product. It has to be said that channels like Social Media can not be the only form of gathering feedback due to difficulties in sampling and focus. A role which focuses on more rigorous testing is necessary – personified by the CL. Closing the circle is the ST which receives continuous information about breaking points and feeds them into the agile cycle.

Regarding **channels and media**: For the POs, a comprehensive project management tool like TFS makes sense and provides an extensible infrastructure for the whole team. For the SMO, Facebook and Twitter as the two largest social media

[1] In the sense of all kinds of communication, not just telephone calls.

are obvious for a mass market product like FinanceX. However, for other projects, the selection of media might be different and include e.g. specialized bulletin boards, depending on the target audience. The fact that the SMO does not use the TFS seems counter-productive. The CLs channels and media are already quite wide (with plans to expand) regarding their recruiting and testing which makes them a good example for other SME interested in incorporating more customer feedback. Cooperation of the CL with the SMO makes sense for recruiting test participants as well as exchanging and discussing impressions. The ST's tools are fairly straightforward, and include chat, phone, e-mail, databases, etc. They seem well thought out and functional. A possible point of improvement would be the utilization of a customer feedback tool similarly to [22] directly into the software which could provide the ST and hence the whole Agile development process with valuable in-situ feedback from the users' usage situations.

The aspect of **filtering** is one of the most interesting and crucial ones we have encountered, yet it is also the most fuzzy one by far, making it difficult to really pin down. One lesson to learn from Mavolio definitely seems to be the necessity for a comprehensive and appropriately widespread net for gathering input as described and discussed in the previous sections. This is not filtering yet but necessary grounding for the filtering process. The next important step is to have structured role distribution regarding the power of filtering and decision making. Mavolio has a grown, divergent structure here: While the POs are clearly the decision makers and the CL and the SMO only gather, comment and subsequently forward feedback to the POs, the ST participates in active pre-filtering.

As explained in the discussion regarding channels and media, CL, SMO and ST all have slightly different perspectives on the customer's feedback and needs and thus, a similar chain of forwarding for the ST as it is practice with the CL and the SMO would seem sensible and pose a possible point of intervention and improvement in the process. This view is supplemented through the differing foci regarding filtering techniques: The SMO leans towards qualitative filtering, being immersed thoroughly in the product and the population of its users. The CL seems balanced between qualitative and quantitative filtering aspects given its scientific methodology. The ST ostensibly tends towards quantitative filtering which is only logical given the throughput rate of a support call center in comparison to the SMO and the CL. Assuming (as was stated multiple times in our interviews) that qualitative and quantitative filtering have to compliment each other, it would seem sensible to engage in pre-filtering as little as possible, respectively as little as sensible: Similarly to the SMOs current practice, it makes sense to match customer feedback with previously made decisions and inform customers accordingly if a (very) similar request has already been denied or is being processed. To do this, a comprehensive database has to be kept and be accessible to all concerning roles – ideally, this would be done in the project management software.

Regarding the operationalization of filtering techniques, quantitative filtering seems to be the more straightforward one – given thorough documentation in database form, customer feedback can be quantified and analyzed rather easily. This data can supply valuable intelligence into trends. However, it seems extremely important to supplement the quantitative view qualitatively: A *good* idea is not necessarily the same as an *often requested* one – valuable ideas and insights can be rare, too. This makes qualitative filtering a necessity. To explain its operationalization with an analogy: In our interviews, we found certain similarities between this kind of filtering and qualitative sciences like ethnography: Deep immersion into and interaction with the product's user base and using the product oneself – getting a *feel* for it and forming *experience* – has been stated as very important and later, different perspectives and their intersections are considered valuable (one could compare this to the concept of inter-coder reliability in qualitative data analysis). Furthermore, a certain distance from possible moderating factors (like budget aspecs or other business influences) seems associated with successful qualitative filtering in a manner not unlike the (artificial) naive approach utilized by ethnomethodologists. All in all, both views compliment each other and if possible, none of them should be viewed on its own when engaging in filtering.

Discussion points and issues

We would like to wrap up our case study with some position points, theses and impulses for discussions which are based on the literature and our case study and which we would like to debate further in the Workshop on the integration of UCD and agile development:

Customer integration is invaluable for UCD: This is the most obvious point and well-established in the scientific community but many SMEs do not seem to accept this yet which should be discussed further.

Agile development seems useful for UCD: Generally speaking, agile seems well suited to be mapped to UCD methods and customer integration through its structure but has deficits (see below) which need to be reflected in a process model.

Multi-stage is crucial: Customer integration at only a single point can falsify the feedback significantly. Those points should be identified and discussed further.

"Listeners" need to be empowered and actually listened to: The people gathering user feedback through the different channels and at multiple stages need to have a voice in the process, e.g. through explicit participation in the agile process and need to be able to challenge decision making processes.

Customer feedback needs to be filtered: Not everything a customer wants can be done or is actually a good idea. Qualitative and quantitative filtering mechanisms should be employed.

Filtering is not easy: Personal needs to be educated, to actually use the product and to develop a frame of mind not unlike that of a qualitative researcher to filter qualitatively. Quantitative filtering needs support systems like databases which should be connected to the project management tool.

The Project Owner is the critical point: The PO needs to make a significant amount of highly UCD motivated decisions which is why she or he needs a grounded (multi-stage) base and related expertise for those decisions.

Consider two POs: Given the previous point, it may be sensible to apply two POs to cover the necessary skill sets and balance the workload. If this is done, it is vital to establish and communicate the different responsibilities of the POs.

REFERENCES

1. Anderson, D. J., and Reinertsen, D. G. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, Sequim, Washington, Apr. 2010.
2. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. Agile Manifesto, 2001.
3. Beyer, H. User-Centered Agile Methods, 2010.
4. Chamberlain, S., Sharp, H., and Maiden, N. Towards a Framework for Integrating Agile Development and User-Centred Design. In *Extreme Programming and Agile Processes in Software Engineering*, vol. 4044. 2006, 143–153.
5. Ericsson, K. A., and Simon, H. A. How to study thinking in everyday life: Contrasting think-aloud protocols with descriptions and explanations of thinking. *Mind, Culture, and Activity 5*, 3 (1998), 178–186.
6. Ferreira, J., Noble, J., and Biddle, R. Agile Development Iterations and UI Design. *AGILE 2007 (AGILE 2007)* (2007).
7. Isomursu, M., Sirotkin, A., Voltti, P., and Halonen, M. User Experience Design Goes Agile in Lean Transformation – A Case Study. *2012 Agile Conference* (2012), 1–10.
8. Lee, J. C. Embracing agile development of usable software systems. *Conference on Human Factors in Computing Systems* (2006).
9. Lievesley, M. a., and Yee, J. S. R. The role of the interaction designer in an agile software development process. *CHI '06 extended abstracts on Human factors in computing systems - CHI EA '06* (2006), 1025.
10. Mao, J.-Y., Vredenburg, K., Smith, P. W., and Carey, T. The state of user-centered design practice. *Commun. ACM 48*, 3 (Mar. 2005), 105109.
11. Nielsen, J., and Molich, R. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (1990), 249–256.
12. Pipek, V., and Wulf, V. Infrastructuring: Towards an integrated perspetive on the design and use of information technology. *Journal of the Association of Information System (JAIS)* (2009).
13. Schwaber, K. Scrum development process. In *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA* (1995), 117–134.
14. Silva, T., Silveira, M. S., Maurer, F., Hellmann, T., Paulo, U. D. S. a., Carlos, C. D. S. a., Carlos, S. a., and Universidade, P. User Experience Design and Agile Development : From Theory to Practice. *Journal of Software Engineering and Applications 2012*, October (2012), 743–751.
15. Singh, M. U-SCRUM: An Agile Methodology for Promoting Usability. *Agile 2008 Conference* (2008).
16. Steria, J. G. W., and Alsos, O. UX and Agile. *Tutorial, NordiCHI conference 2010* (2010).
17. Strauss, A., and Corbin, J. *Basics of Qualitative Research*, vol. 3. 2008.
18. Sutherland, J., Schwaber, K., Scrum, C.-c. O. . C. O., and Sutherl, C. J. The scrum papers: Nuts, bolts, and origins of an agile process.
19. Sy, D. Adapting Usability Investigations for Agile User-Centered Design. *Journal of Usability Studies 2* (2007), 112–132.
20. Wharton, C., Rieman, J., Lewis, C., and Polson, P. The cognitive walkthrough method: A practitioner's guide. In *Usability inspection methods*, John Wiley & Sons, Inc. (1994), 105–140.
21. Williams, L., and Cockburn, A. Agile software development: it's about feedback and change. *Computer 36* (2003).
22. Yetim, F., Draxler, S., Stevens, G., and Wulf, V. Fostering continuous user participation by embedding a communication support tool in user interfaces. *AIS Transactions on Human-Computer Interaction 4*, 2 (June 2012), 153–168.