User Integration in Agile Software Development Processes: Practices and Challenges in Small and Medium Sized Enterprises

Oliver Stickel[1], Corinna Ogonowski[1], Timo Jakobi[1], Gunnar Stevens[2], Volkmar Pipek[1], and Volker Wulf[1]

[1] University of Siegen, Germany
firstname.lastname@uni-siegen.de
[2] Bonn Rhein-Sieg University of Applied Sciences
gunnar.stevens@h-brs.de

Abstract HCI and CSCW research as well as practice has strongly indicated the value of integrating (end) users in software development processes. Such integration can help address actual needs and wants, to avoid undesirable developments and to strengthen the User Experience of a product. A user-focused approach to software development has some conceptual overlap with agile software development practices, such as quick and iterative (user) testing. However, out in the wild, organizations seem to have difficulties actually mapping user centered development with agile processes for a variety of reasons ranging from organizational or hierarchical aspects up to financial issues. This problem seems specially prevalent in Small and Medium sized Enterprises (SME) where such constraints can be even tighter than in larger organizations. To help understand those problems and to identify possible solutions, we turned to three quite different German software SME, varying in size, market focus and organizational structure. By way of qualitative field studies, we were able to identify key roles and tools as well as methodological, organizational and analytical practices and challenges in integrating (end) users into agile software development.

Keywords: Agile Software Development; User Centered Design; User Feedback; Case Study; Qualitative Study

1 Introduction

Software has become an invaluable part of private and professional life allover the world. We know that this leads to Usability and User Experience[3] becoming more and more important factors for the success or failure of ICT systems. This obviously holds true for all sort of systems which can be interacted with by

[3] From here on out, we will abbreviate "Usability and User Experience" as UUX. For the purprose of this contribution, we do not need the distinction between more task-focused and more ludic aspects.

users, but for the purpose of this contribution, we will focus on software systems. For the software world, we have solid research [13] as well as norms such as the DIN EN ISO 9241 supporting the integration of (end) users in all phases of a development project as one of the most central factors for positive UUX.

If we look at the economically important sector of Small and Medium Sized Enterprises (SME), however, we seem to find indications for deficits in actually focusing on UUX. Hering et al. [14] indicate aspects such as financial, logistical, hierarchical or methodological issues as factors holding the SME sector back in regards to the systematic integration of users and user feedback in development processes. Furthermore, the norms and process models such as the aforementioned ISO 9241 or User Centered Design (UCD) often lack clarity in regards to the *actual* implementation of user integration and, equally important, its integration with established process models in organizations.

Our contribution is situated right in this research / practice gap. Its goal is to help identify the actually relevant issues for SME when dealing with user integration as well as to assist in finding solutions. We base our work on a practice-based, socio-technical understanding of Human Computer Interaction (HCI) and Computer Supported Collaborative Work (CSCW) [34]. Consistent to this base, we chose qualitative case studies in three contrastive German software SME as our main research instrument. We worked with the following organizations for our field work:

Foo[4] is one of the largest German SME in the software business focused on end users with quite nuanced processes for the integration of user centered methods and agile development processes.

Bar is a relatively large SME (if decidedly smaller than Foo) producing soft- and hardware for end users. Bar's focus on UUX has a briefer history and smaller extent than Foo's.

Qux is a very small, design-driven software company which mainly fulfills orders, i.e. with no direct end user market.

Based on our fieldwork in all three organizations, we were able to identify *Roles*, *Channels and Media* as well as *Interpretation and Filtering* of user feedback as the three main categories moderating (and moderated by) the success or failure of user integration in agile development processes. In the following sections, we first give an overview about the relevant state of the art before reporting our results and discussing them with a focus on the three main categories.

2 Related Work

In this section, we will present an overview about the relevant scientific background and literature, starting with a very brief primer on agile software development,

[4] All organization names as well as all personal names which will follow in later sections of this contribution are anonymized for privacy reasons as per agreement with partner individuals and / or organizations.

leading up to the relevance of user integration for positive UUX and finally the synthesis of both aspects.

2.1 Agile software development

Agile software development [2] refers to relatively new paradigms to structure ICT projects. The most common agile schools are Scrum [26] and Kanban [1]). Agile methods differ differ from classical process models such as the "waterfall" 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 always be change and propose to prepare for this and embrace it [32]. 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.

2.2 User integration for a better UUX

Existing literature does motivate the integration of users and customers in the design process for several reasons, including UUX as well as political, economical and ethical ones [35]. Participatory Design (PD) as maybe the earliest systematical approach for user involvement originates in workplace democracy movements (e.g. [9,4]), supported by trade unions. However, commercial software companies implemented PD and its methods such as collaborative storyboarding or Group Elicitation approaches as well [11]. *Active* user participation was deemed to be effective since the actual users of a product know their own perspectives and needs best [24]. UUX is not necessarily the explicit core focus of most "original", political PD approaches. However, the American school of PD dropped the political framework and argued mainly for efficiency of the created products [16], taking user integration in a more UUX-focused direction.

Later approaches to user integration in ICT development projects include Integrated Organization and Technology Development (OTD) [33] as well as STEPS [12]. Both approaches are normative software development models, that involve close participation of users and developers. Both are also more focused on their application in organizations. However, as they are designed for a very close and rather intricate collaboration of developer and customer organizations, they are not easily usable for the development of mass-market off-the shelf software, especially not for SME with limited ressources [13]. Looking at the earlier stages of ICT projects, we also should mention von Hippel (e.g. [15]), who has long focused on User Driven Innovation and its benefits. However, the focus on early phases also frame the limits of this approach.

The current trend of user integration in ICT projects is often framed referring to User Centered Design (UCD). UCD, as codified in ISO 9241-210 (more e.g. in [23]) can be seen as a normative design and development model that argues for user integration in all phases of the process. The UCD ideology specifically views the user as an asset of the product development process. Furthermore, unlike older models, it explicitly focuses on generating a positive UX as well. 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.), to optionally include users in mock-up generation or evaluation and finally include users in the evaluation of releases. UCD 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.

2.3 Synthesis and research gap

It has to be noted that Agile methods already seem to incorporate some aspects and resemblances to UUX approaches which focus on the consumer. Essentially, at this point, we are looking at the room between design and engineering teams left by the UCD, we talked about in the last section.

The iterative approaches of agile development and the emphasis on the customer [2] are, at first glance, quite compatible with UCD [6]. There are multiple positive reports on adaption and integration attempts, e.g. a single case study on a multinational coorporation and its shift to agile [17] or [31], a case report from another big corporation 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 UUX of their products. However, it has also been noted quite frequently (e.g. [20,10,22]) that the actual integration work of UCD, and more generally, different UUX methods, is often not optimal in practice and is not yet well understood. Hence, there have been different scientific workshops and tutorials, e.g. [29,19] as well as suggestions for procedural models or frameworks to facilitate the integration. [27], for example, base their framework on an Interaction Design Lifecycle and specifically include design cycles into the agile process models as formal institutions. [3] focuses on UUX professionals and how to integrate them in Agile environments, not least by facilitating understanding for such development strategies on the UUX side. The Scrum roles themselves are also regarded as relevant for the successful integration of UUX and Agile. [28] identifies the Project Owner 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 often not qualified in-depth for UUX. This leads them to the proposal to appoint two POs, one of which focuses on more traditional responsibilities in the Scrum model while the other ones' responsibilities lean towards UUX [28].

All in all, the state of the art implies that there are a great many and topical relations between UCD and UUX methods and agile software development. There also have been investigations and practical attempts at the integration of both approaches leading to some beneficial results in practice as well as some more theoretical concepts on the integration. However, prior work strongly indicates that the understanding of UCD/UUX and agile still leaves many gaps, especially in regards to the systematic understanding the actual practices and challenges faced by organizations *in the wild* [8,10]. This contribution is an attempt at helping to fill this gap by way of three comparative case studies with a focus on the domain of SME.

3 Cases and Method

In the following sections, we will first describe our three cases in more detail before explaining our encompassing methodology and our analytical process. Subsequently, we will then give an overview about our data and coding scheme.

3.1 Foo – a very large SME with established UUX practices

Foo is a very large SME with roughly 500 employees and a strong corporate culture towards UUX and user integration. Their product focus is centered on software systems for end users with an emphasis on personal and organizational finance administration and management tools. We mainly worked with one project team within Foo which is responsible for iFin, a tool for personal finance management. iFin is a mass-marked product, cross-platform (mobile and desktop, multiple operating systems) and its development is agile, utilizing Scrum.

At Foo, within the iFin team, we conducted semi-structured interviews of about 60 minutes with the responsible Product Owner (PO), the Social Media Management, the head of the support team (for all products, not just iFin), a member of the support staff specialized on iFin, a developer as well as two members of the in-house usability lab. Those interviews were supplemented by multiple later rounds of further inquiries to the interview partners via Skype, phone and mail. We also conducted participant observations during usability tests in the in-house lab (3h) as well as during a Scrum planning meeting (4h).

3.2 Bar – a big SME with emerging UUX practices

Bar is a big SME with about 200 employees. Bar has focused on customer home electronic components for a longe time. Especially in the area of home network components, Bar has a developed nuanced and processes and competences. However, more recently, Bar decided to develop a line of Smart Home components which was starting out on the the market at the time of our study. This also led to more focus on interface design and development as well as UUX due to the bigger amount of user interaction with smart home devices in comparison to more passive network electronics. Our work with Bar focused on the smart home team and their developing, emerging agile development processes as well as their in-house user test sample for working with prototypes.

We conducted semi-structured interviews of about 90-120 minutes with the heads of the development team, the team for Design, Verification and Testing (DVT), the product marketing as well as the responsible PO.

3.3 Qux – a small, design-driven software agency

Qux is a growing but still small software development agency with 11 employees. They offer development and consulting services as well as design of innovative software, mobile apps as well as digital products in areas such as Internet of Things and e-Mobility. Being an agency, Qux' focus is less on selling to end users directly but rather on delivering on orders from other organizatioins. Qux has a corporate culture and image with a strong focus on design-driven development and decision making.

At Qux, we conducted semi-structured interviews of 60-90 minutes with two of the three of the firm's partners which also act as Scrum Master / Head of Project Management (PM) as well as Creative Director (CD), respectively. Similar interviews were conducted with a Senior Art Director UI/UX as well as a Mobile Developer.

3.4 Methodological framing and analysis

How does [Foo / Bar / Qux] integrate user input and feedback into their agile software development process and how does this relate to UCD?

This was our quite basic research question motivating our study. It is important to note that we did not approach the field theory-driven but rather based our approach on open, field-driven research, inspired by Grounded Theory (GT) [30]. Therefore, in each case, we quickly went into the field where we iteratively developed our understanding of the companies' practices and our research strategy according to our findings. We deemed a (field-)data-driven approach to be important given the disparities between theory and practice and the ambiguities described in the state of the art.

In total, we conducted 15 interviews. Seven of them were at Foo, four at Bar and Qux, respectively. The interviews lasted between 60 and 120 minutes and were recorded as well as transcribed pragmatically, meaning full transcriptions utilizing only markers for salient events such as laughter, peculiar facial expressions or breaks. However, we did not include micro-expressions, precise counting of break time intervals, detailed pitch analyses, etc. since we did not deem such data necessary for our research interest in practices and challenges. All interviews were semi-structured, utilizing a guideline which evolved in the field, led by the field. The interview language was German in all cases, the quotations in this contribution are translated. Transcripts were supplemented by handwritten field notes and memos (about 25 pages). Furthermore, we gathered artifacts such as

User Stories, bug reports or usability reports, mainly at Foo because of its bigger size and availability of many such artifacts as well as due to the fact that Foo was our first case study and the data helped us to open up the field. Finally, we supplemented our interviews by multiple further inquiries to the interview partners via Skype, phone and e-Mail during the analytical process each time relevant questions arose. Brief descriptions of the data sources can be found in sec. 3.1, 3.2, 3.3 and an index in tab. 1.

All data and artifacts were subsequently coded. In GT-speak, we coded axially and selectively [30]. However, we do not claim to have established a 'Theory of UCD and Agile' – we feel that such and encompassing theory would necessitate multinational and even more contrastive cases as well as a longer period of time. However, our analytical process followed GT methodology and can serve as one of many pieces in a more comprehensive puzzle towards a theory. To clarify even more: One might also say that we oriented ourselves on Thematic Analysis (TA) [5] which, essentially, is GT without the overhead towards extensive theory building but with the option to add that on top iteratively. The coding process started immediately after the first interview and was continued and evolved all throughout the research activities. During the field research phases, we held weekly discussion and mirroring meetings in regards to the the coding activities in our research group. This also included researchers which were not active in the field, some not even in our research project at all. Those researchers helped asking questions the field researchers were not thinking about given their different perspectives, forcing the latter to explicate a significant amount of tacit information. The permanent condensation of the coding structure was followed up to the point where the gathered data did not add significant new insight (saturation). This also helps explain the different scope of data between the three cases – with the evolution of a denser coding scheme, (transferable) insights led up to the saturation points more quickly as per the intention of GT-inspired approaches.

3.5 Data overview and coding scheme

In tab. 1, we have indexed all interviews and observations. For clarity: I-F-04 was an interview with two participants (the full staff of Foo's in-house usability lab), I-F-07 was an interview with a PO for a different product team than iFin since this PO was referred to as as one of the central experts in regards to agile software development and UCD in the company[5] and in cases such as I-B-03 and I-Q-01, one person fills multiple roles. In tab. 2, you will find an overview about the very basic structure of each participating organization and our central coding scheme which boils down to the three codes "Roles", "Channels and Media" as well as "Filtering and Interpretation". Our report which will follow in the next section will be oriented on this structure as well.

[5] At this point in the analytical process, it had already become clear that the intersection of those two topics would be central to our study.

ID	Description	ID cont.	Description cont.
I-F-01	Product Owner iFin	I-B-01	Product Owner
I-F-02	Social Media Officer	I-B-02	Head of Development
I-F-03	Chief of Support	I-B-03	Head of Design, Verification & Testing
I-F-04	Customer Lab (2 empl.)	I-B-04	Head of Marketing
I-F-05	First Level Support	I-Q-01	Company Partner 01: Scrum Master & Head of Project Management
I-F-06	Software Developer	I-Q-02	Company Partner 02: Creative Director
I-F-07	Product Owner (other project)	I-Q-03	Senior Art Director UI / UX
O-F-01	Sprint Planning	I-Q-04	Mobile Developer
O-F-02	Two Usability Tests		

Table 1. Index: Interviews and Observations

	Foo	Bar	Qux
Size	about 500 employees	about 200 employees	11 employees
Product Portfolio	Variety of software around finance administration.	Home network equipment. Recently, soft- / hardware for the smart home.	Software agency with a wide variety of projects, mainly for corporate customers.
Study focus	iFin, a cross-platform personal finance management tool.	iHome, a soft- and hardware ecosystem for smart homes.	No specific project, lateral study through the company.
Code: Roles	"I answer requests myself, however if they are very technical in nature, I ask the support people [...]" (I-F-02)	"Since the DVT is our last line of defense, they have to check somehow what has been developed..." (I-B-01)	"I think we are quite good in putting ourselves in those roles [users] [...]." (I-Q-02)
Code: Channels & Tools	"We use several [channels]. We mainly work with two platforms: Facebook and Twitter. Also, we maintain a blog for iFin" (I-F-02)	"If somebody had an issue beyond hard problems, they did not necessarily put it in [into Bugzilla]." (I-B-01)	"[...]we meet at 9:00 and everybody explains what he did the day before and what he plans to do today..." (I-Q-03)
Code: Filtering & Interpretation	"If features are requested [by users] that are very special, we can assess if this is interesting for the masses or not." (I-F-01)	"I give the thing [iHome] to 10 people and get 10 different opinions when I ask a specific question." (I-B-01)	"For the most part, it is not very good if the customer selects [test] users itself but it's just the way it is." (I-Q-03)

Table 2. Organizational and Coding Scheme

4 Results

In this section, we will report on the three most important themes as listed in tab. 2 as well as their interrelations, starting with the *roles*, leading up to *Channels and Tools* and finally, aspects of *Filtering and Interpretation.*

4.1 Roles

Foo: With Foo, we saw that a multitude of roles is in contact with users. The support team obviously has most points of contact since they are confronted with a wide variety of user issues on a daily basis. However, they are not only trained to solve those issues but also to try and understand where they come from and ask for more feedback than strictly necessary to solve the issue in order to provide input for the product development. To this end, Foo has kept their support team in-house, located near to the development, management and other teams as well as actively trying to foster a culture of deep and long-term engagement with as well as knowledge about Foo's products. To let a first level support employee speak for himself:

> *"I've been working for Foo for about ten years now. I can use the software blindly. I can find problems and difficulties while standing on my hat."*
> I-F-05

Foo also has an in-house 'customer lab', which is a traditional usability lab, staffed with two UUX-experts. They carry out structured user testing on request by the development teams and report to the Product Owner. Furthermore, Foo also has a defined role for Social Media management (SMM). The SMM constantly tries to engage with users by way of providing them with information, monitoring discussions, trying to mediate if necessary and very consciously tries to get a "feeling for the mood" (I-F-02) on Social Media in regards to Foo's own products as well as the competition's.

As in established Scrum doctrine, we found the role of the PO to be the central hub within the different weaves of user integration and user contact in Foo. There are two notable observations in regards to Foo's PO structure for iFin: First, there are actually three POs: One is managing daily affairs such as codifying user stories, the second one is focusing on the epics and the third one has a background in design. The PO-team's skills compliment each other, however, they also consult with internal experts (such as the SMM) on a case-by-case basis. Second, while it certainly is in Scrum-spirit that the PO represents users (and hence, has to engage with them as well), some of iFin's long-term users even have the POs phone numbers and call them occasionally, especially when something in the product changes in a way they do not like.

Software developers themselves do not have user contact in Foo.

Bar: Bar has a product-oriented organizational structure based on Business Units. Within the unit we worked with, direct user contact is limited and structured in clear channels through a fixed test sample of users as well as internal testing (more on this below). Bar's division 'Design, Verification and Testing' is the central role responsible for testing and validating new software releases in regards to bugs and completeness compared to the requirements:

> *"Since the DVT is our last line of defense, they have to check somehow what has been developed . . . Meaning they always put the requirements next to the result [result = a release]"* (I-B-01)

At the time of writing, Bar also established a support team structure intended to closely work with users with the explicit goal of feeding back into the product development. Like Foo, Bar also utilizes Social Media as well – however, with Bar, Social Media work is co-located within the marketing division, whereas Foo has a separate, explicit organizational role for Social Media management. Bar's development team is more distributed than Foo's, including more external partners, and with the in-house development team focusing more on coordinating and conception. Like with Foo, Bar's central role for UUX is the business unit's one PO. He handles all reports and user feedback and makes all decisions in regards to UUX:

> *"[...] Then, they [user feedback and feature requests, consolidated by the DVT] came to me. [...] and I had to go back to the Wireframe or make clear how this and that is intended [...]."* (I-B-01)

However, he consults with external agencies on a case-by-case base. The PO has direct user contact, mainly in case of concrete, deeper enquiries and user problems within the test sample.

Qux: Qux has a very flat hierarchy which splits up in a Design- & Development unit, Social Media Marketing as well as the roles of both company partners who act as Scrum Master & Head of Project Management (PM) and Creative Director, respectively. The PM is responsible for internal quality control of concepts and releases while the CD and the Senior Art Director UI/UX (AD, located within the Design- & Development unit) manages all UUX aspects. Hence, Qux has formed a structure where the PM takes on what might be called the more managerial aspects and the AD the user-focused ones of what Foo and Bar subsume under the role of their respective POs.

A central distinction of Qux as a software agency in comparison to Foo and Bar is that their customers are generally not their end users. Hence, we find roles such as support teams and product-specific Social Media engagement not within Qux but rather within their portfolio of customer organizations. Wider user tests (and hence, roles with user contact) are also either sourced out to third parties or the respective customer organization takes charge of those activities itself. Qux also actively asks the customer organizations for feedback after each sprint.

Furthermore, this structure brings with it a certain fluidity of roles: On a case-by-case basis, Qux leverages all of its staff as well as friends and family for ad-hoc testing and feedback. This culture is illustrated quite well by the actual Senior Art Director UI/UX:

> *"[...] my father who has no affinity for such things [ICT]. . . I really like to just hand him stuff [beta versions] – just to look at what he does." (I-Q-03).*

The dynamic feedback loops between roles and units all converge at the PM. This approach of internal testing is based on a quite explicit corporate culture focused on User Centered Design, encouraging the staff to constantly take a step back and actively try to view the product through a customer's as well as a user's eyes:

> *"I think we are quite good in putting ourselves in those roles [users] [...] When somebody is working on a project, we also try to put him together with a colleague working on a different project [...] to get a different view. I think that's really important."* (I-Q-02)

4.2 Channels and Tools

Foo: Central to Foo's agile Process is Microsoft's TFS which gets used as a code repository as well as for handling and prioritizing artifacts such as bug reports, feature requests and usability test reports. Especially the developer team as well as the POs utilize TFS constantly to manage and track iFin's development.

The support team utilizes E-Mail, phone, fax, letters, chat as well as product-specific web-forum to engage with users directly, although the forums are focused on a *"customers help customers"* (I-F-03) approach. User feedback gets taken from the support-specific ticket system and is put into the TFS if deemed valuable (more on this distinction in 4.3).

The Social Media management mainly utilizes Facebook and Twitter and, to a smaller extent, Blogs as channels to interact with and include users. Notably, she does not use any special Social Media management tool. She tries to contextualize user feedback as much as she can utilizing the rich data provided by Social Media. Subsequently, she directly engages with the POs in regards to the feedback utilizing e-Mail or face-to-face conversations. It is notable that she does not use the TFS even if she has access to it. Furthermore, regular surveys utilizing the Net Promoter Score [25,18] are carried out. In case of problems like server outages, known bugs or similar issues, the SMO informs the customers over the available channels and, more importantly, keeps them up to date. An example from I-F-02 was a bug occurring after an update which crashed the app immediately after starting it. A bug-fix was implemented and submitted to the app store very quickly but due to the approval process in the concerning store, the update needed time to be made available to the customers. The SMO kept the customers informed every step on the way which received quite positive feedback.

The Customer Lab's main channels and tools are traditional user tests with Thinking Aloud and sometimes Heuristic Evaluations and Cognitive Walkthroughs, although they branch out towards methods such as Contextual Inquiry-inspired approaches, even at users' homes. The CL utilizes usually utilizes series of tests with 5-20 participants and frames the results as comprehensive reports in a structured format. These are subsequently put into the TFS for the POs. Notably, it is also possible for everybody in the development team to tune into live video feeds from the usability testing sessions, although it has been expressed in I-F-04 and I-F-06 that developers usually do not do this, stating that the *"reports are enough"* (I-F-06). Tests in the CL are only carried out on request of the POs, the management or other decision making roles.

Apart from the TFS, the POs also have product-specific e-Mail accounts for free-form feedback which can be reached by the users from within iFin. Furthermore, the POs actively monitor as many app stores and similar places in the Web where users leave feedback of some sort. Foo even developed an in-house tool, specifically for the purpose of aggregating such reviews and making them manageable. As mentioned before when we tried to illustrate the self-conception of the involved roles, lead users also sometimes contact the POs in person utilizing phones as well as by e-Mail. Foo's POs also receive a certain amount of automated use tracking data. However, this is kept to very specific and intensely debated cases due to privacy concerns. In line with Scrum doctrine, one of the most central tools for Foo's POs are User Stories which are built, maintained and utilized without any company- or project-specific peculiarities.

All interviewees in Foo talked about the importance of *"Flurfunk"* (I-F-01) (literally "corridor radio"– office grapevine), coffee corners and informal meetings for coordination, sharing and discussing user feedback and user perspectives.

Bar: Bar utilizes a custom in-house database system geared towards product management in which all requirements and properties of the product are being held and maintained. Great emphasis is placed on a set of wireframes of the final system. Those wireframes have been developed quite early in the development process of iHome and they can be understood as similar to traditional target specifications for internal purposes as well as in a sense of coordinating artifacts with external contractors:

> *"[. . .] Meaning they [the DVT] always put the requirements next to the result [a release] and can refer to the wireframe [. . .] [interviewer asks about how the wireframes changed during the development process] Well, they stayed relatively stable in scope [...] here and there, there were adaptions [. . .]* (I-B-01)

With Bar, systematic user testing is only being applied after all desired features as specified by the wireframes have already been implemented. The focus in this phase is not on finding innovative new features or investigating into end-user appropriation but rather to purposefully shape UI-components and interaction flows. To this end, Bar has joined forces with an external partner in order to establish a Living Lab [21] infrastructure as a test bed: About 30 households get the product a few months before rollout in order to test it in their homes. Those tests happen without instruction or rules apart from a commitment to actively use the system and test specific features after updates. Bugzilla has been implemented as a channel where users can put in tickets. This is supplemented by occasional informal exchanges face-to-face. During the Living Lab phase, Bar also recognized that users seemed not to input all their problems into Bugzilla, especially when the problems in question did not relate to hard and evident bugs:

> *"[. . .] If somebody had an issue beyond hard problems, they did not neccessarily put it in [into Bugzilla]. There are many kinds of problems [...] like nice-to-haves, problems with understanding things or other issues like that.* (I-B-01)

Additionally, comprehensive automated logging about use data is conduced in the background with the goal of making issues reproducible. After commercial rollout, Bar's plans are to have the support as well as the marketing divisions report directly to the PO about user feedback.

Qux: Qux uses Jira and Confluence as a basic infrastructure in order to scaffold agility in their development process. Those tools are utilized for internal coordination, especially for the PM. Furthermore, Qux' intention is also to establish customer-facing transparency. Hence, customers can also input tickets and feedback (depending on the project structure agreed upon with the customer).

In regards to active user feedback and participation, Qux employs different methods. In some cases, customer organizations carry out their own beta testing and feedback gathering, aggregate it and send it to Qux. In other cases, all data from such tests is handed over to Qux without aggregation. Another option is relying on direct user feedback via e-Mail generated from feedback-buttons and similar options integrated into applications, without the involvement of customer organizations. Qux' employees are aware of a wide variety of tools and systems to facilitate user feedback such as Testflight or crowdsourcing systems, but on various occasions throughout the interviews, it becomes clear that they are still searching for an optimal system, especially one that meshes with agile development:

> *"[. . .] In each release in Scrum, there is one functional area, which gets completed and released, so to speak [. . .] there's always this wish, we are looking for a suitable platform [. . .] so we can say: 'you don't have to send me an e-Mail, you don't have to write down anything, you don't have to call me [. . .] then they could just hit a button, rate it [the specific*

> *result of a sprint/release], write a short text, Twitter-style at most [. . .] which would then just be sent to us so we could look at it."* (I-B-01)

Apart from users, the customer organizations themselves also actively queried as sources for feedback. Qux' PM puts it like this:

> *"[. . .] we obviously also collect feedback from our customers. When we present something [. . .], we ask them quite focused: '[...] please look at this'. We have them take responsibility, which is a good thing, since basically, it is their project. . . Which is why I expect them to care and not just complain in the end, after a release [. . .] They have to give feedback frequently.* (I-Q-01)

Furthermore, Qux frequently employs app store reviews and ratings as feedback channels, similarly to Foo. For more qualitative evaluations regarding UUX and UI, Qux has no formal tools or channels in place. Here, they rely on a user-focused and agile company culture as described above as well as ad-hoc feedback in meetings with customers and beta testers. Internally, user/customer and/or peer feedback is not just shared via Jira but especially utilizing Daily Stand-Ups which are emphasized as an important tool:

> *"[. . .] we meet at 9:00 and everybody explains what he did the day before and what he plans to do today [. . .] you don't put things off[. . .]"* (I-Q-03)

This ritualistic form of informal exchange is supplemented by grapevine, similarly to Foo's case. Like Foo, Qux then utilizes User Stories and Bugs in Scrum-fashion to codify and work with the user feedback. Both artifacts are primarily put in and maintained by the PM. To be clear: The PM is not the only *source* of such data, as explained above, but he is *maintaining* it.

4.3 Filtering and Interpretation

In the previous sections, we tried to report on how and with the participation of which roles customer feedback is gathered and passed along through Foo, Bar and Qux. There is, however, one step missing in between – what emerged as *filtering and interpretation*. This is, in a nutshell, the process of analyzing and assessing user feedback as well as matching it with other feedback and/or internal goals. It also encompasses on the challenge of identifying what the user *really* means or needs.

Foo: Foo has a long history of experimenting with the incorporation of user feedback in their development cycles and within this history, there have been failures, too. One example from the interviews (I-F-01, I-F-03, I-F-05) is a former project grounded in Participatory Design where the development of a software product relied heavily on the input of a selected group who were considered lead users in their. However, as it turned out later, the product became much too specialized and thus did not appeal to many potential customers. Experiences like this reinforced Foo's focus on filtering as well as diversifying user feedback structures – 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. What this means is that in the example of iFin, which has a very widespread and heterogeneous user base, specialization has to be kept at a much broader and shallower level than for some of Foo's other products, e.g. those targeted at landlords and this much smaller and more focused group's specific needs.

The most important decision makers regarding the filtering process are the POs. They consciously try to match their vision of the product with the customer input, adapt, prioritize and, if deemed necessary, modify or reject specific feedback. The customer lab does not engage in filtering per se but rather reports comprehensively and based on proven methods. The Social Media management engages in partial filtering – she ties to match every piece of incoming input with previous decisions made by the POs and if the input is identical or very similar, she *"informs the customer accordingly"* (I-F-02), meaning information such as 'request denied', 'request in development' and so on. In such cases, she does not alert the POs. If a specific piece of user feedback is new and she hands it over to the POs, she usually annotates it and states her opinion about it, i.e. actively enriches the user feedback based on her long-term experience with iFin. Notably, the support team does filter actively. Customer feedback is gathered, discussed between the leader of the support team for the respective product and the chief of support and filtered. Thus, some feedback may never even reach the POs:

> *"If a person wants a new feature, the support employee checks the database weather the feature was already requested by someone else. If so, the customers ID is added to the incident. If not, the request is inserted in the database, which triggers a message to the team leader who assesses it. If he decides that the request is useful, the entry is set to 'visible' for the PO and the development team." (I-F-03)*

Throughout all interviews with Foo, the exact operationalization of the filtering processes remains somewhat vague 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'[6] 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. However, this feature would make other, quite specific, long-term plans for the software impossible on a technical level. Hence, it is not implemented. The opposite constellation,

[6] Terminology taken from the interviews – 'call' should be understood as all kinds of communication with users, not just telephone calls.

i.e. individual or occasional cases of feedback 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 a certain 'artfulness' rather than just hard data. It was next to impossible for all interviewees to really describe techniques and methods for qualitative filtering. Instead, in nearly every interview, it was stated ex- or implicitly that a *lot* of knowledge about and a *"feeling for"* (I-F-01, similarly phrased also in I-F-03 and I-F-05) the product has to be developed over time in order to 'get it right'. A need to actually be a user of the product oneself has also been mentioned. All in all, Foo's Social Media management is the role with the deepest engagement in qualitative filtering procedures.

Qualitative and quantitative filtering mechanisms are reported to compliment each other well, e.g. within I-F-01 and I-F-04 and none of both aspects is viewed as sufficient by itself.

Bar: Similarly to Foo, Bar's PO has a key function in the process of filtering and interpreting user feedback. He classifies and judges incoming information and notes from a range of roles through different channels. He has to decide and to match the pieces of information with the long-term goals for the product. On a quantitative, heuristic level, Bar judges feedback as relevant if it comes in 2-3 times in similar form:

> *"I give the thing [iHome] to 10 people and get 10 different opinions when I ask a specific question. That's rather difficult. At the moment, my strategy is that I look deeper into things after I hear issues 2-3 times. [...] Well, I always look at all the things [feedback], but when A says A, B says B and C says C, I stay with my opinion."* (I-B-01)

However, Bar's PO is also aware of the pitfalls of such an approach:

> *"But usually, you feel a bit like, well, the father of such a system. That makes each [...] feedback which is not precisely like you think about the system critique and you have a certain defensive position. It is difficult to be neutral."* (I-B-01)

Filtering and categorizing feedback coming in from Bugzilla is managed by the DVT. They decide when and if something gets bumped up to the PO for decision making or directly to the development team for implementation. Our interview partners at Bar stated that informal, ad hoc (coffee corner-)talks between DVT, developers and the PO are central instruments in discussing, judging and triangulating feedback. Hence, we can also categorize filtering and interpretation mechanisms in Bar in quantitative and qualitative aspects. However, the structures are less complex and less differentiated than in Foo's case.

It is notable that Bar has yet to establish formal structures in who actually takes responsibility for the liaison and engagement with their test households, leading to difficulties in regards to filtering and interpreting feedback coming from those households:

> *"[. . .] there is the question if the developer should have frequent contact? I just don't know. Partially, sure, so he can hear opinions face-to-face and hear users' problems – just to understand. [...] [the users] all have their opinions. That has to be channeled in some way. Can you categorize such things?"* (I-B-02)

Lastly, Bar's PO is unsure if the test users might not become blind to certain issues due to routine and debates if the user sample should not be constantly changed, at least partially:

> *"[. . .] you just breeze over certain issues [after engaging with the product becomes routine] [...] if the beta-tester is at the point [where an issue arises] for the second time, he just skips it [referring to ignoring issues or finding workarounds]."* (I-B-01)

Qux: The situation for Qux is quite different than for the other two SMEs in regards to the filtering and interpretation of feedback. Through their role as a service provider, they face the dichotomy of having to engage and negotiate with their customers as well as having to discuss and judge user and customer feedback internally. Qux' flat hierarchy is helpful in providing a lean and agile structure to quickly engage with such feedback. The strings of such decision making processes all converge on the PM but subsequently have to be debated with the customer who makes the final calls, sometimes putting Qux in less than optimal decisions:

> *"For the most part, it is not very good if the customer selects [test] users itself but it's just the way it is; we don't have the target audience on board. It's a shame but it's the way it is."* (I-Q-03)

Yet, Qux' employees voice rather unequivocal support for User Centered Design, UUX and customer integration and – as mentioned before – reflect on those topics frequently. The Creative Director puts this in simple, decisive words:

> *[Question about what would speak against a strong UCD-motivated process]"Just ignorance. If you do UCD, you put the user or the user group in your focus [...] which is logical. We don't build things for animals or for little grey men but for people."* (I-Q-02)

Given their company structure, current practice in Qux' development process is utilizing as many automated use tracking and data gathering tools as possible since those can be integrated easily and quickly (and cheaply) into their products. This quantitative feedback is then triangulated with qualitative information which mainly comes in through e-Mail, either directly from users or aggregated from customers. This practice meshes well with Qux' agile process. Quantitative heuristics similar to Bar's case are employed to speed the process up but Qux' sentiment is that qualitative information gathered by direct exchange and engagement with users would be more valuable. However, limited staff resources as well as the Qux-customer relationship make this hard to implement and also limit the options for Qux' staff to pose questions to users if they arise about qualitative data aggregated by Qux' customers. Trust in the validity of such data is critical but Qux has no first-hand way of ensuring this.

5 Discussion

Based on the three quite contrastive case studies, we can see that the amount of users a company can engage with as well as the differentiation of channels and tools can scale with the company's size. This is problematic since small companies like Qux which perceive a need to engage more with users simply cannot do so adequately. It is certainly possible to utilize internal testing, using ad-hoc methods such as convincing friends to give feedback or put a mental emphasis on a user perspective – however, it seems that the more people actually try to do this, the more they realize that such methods might be inadequate and the pitfalls such as too much introspection or blindness to certain aspects are many. Such concerns become especially obvious when a second web of entanglements – external customers – becomes part of the process. A possible solution to such problems might be working with external, specialized partners for certain aspects of user engagement, such as Crowd-Testing platforms or testing *as a service*[7]. Through economics of scale, such services can be offered more cheaply than building complex infrastructures for user engagement and feedback internally and might be an entry point for more in-depth work with users, as witnessed by Bar's case.

The differentiation of roles is a highly interesting factor. It, too, can – and maybe even has to – scale with the size of the company itself. With Foo, we have an exceptional example where, over the course of many years, a very intricate web of different roles has emerged. Those roles and their different perspectives compliment each other well and eliminate many of the insecurities and problems we saw in the other cases, such as Bar's issues with responsibilities for certain aspects of working with users or Qux' problems with data validity.

However, different roles do not just magically compliment each other – they also clash and Foo shows how to facilitate this in a purposeful manner. Company culture is the keyword here – Foo's Social Media management might, for example, disagree *very* strongly with a PO's vision for something since she actively tries to take on qualitatively grounded user perspective. Heated discussions can happen – and according to Foo, they should. All roles need to be empowered

[7] An example might be Living Labs as a service, see e.g. [21].

enough not to fear personal or other negative consequences yet still be able to get behind the overall product vision. Agile development can help here because it facilitates constant, quick engagement within teams and provides structures in which things can be explored and tested without great risk or cost (and hence, usually personal consequences). UCD can help as well because a common denominator in developing products *for users*, not ’for oneself’ can put things into perspective, especially in regards to company culture. In Qux’ case, we see that even a small company can form a very strong user focus in its culture. We also see that this has significant impact on the products (if not as much as a combination with a strong base in resources and differentiated roles, tools and channels).

Despite the importance of a careful mesh of roles, tools and channels, there seems to be one central point which can make or break user centered agile software development and that is the PO. There is a very great deal of power about the agile process itself as well as about UCD/UUX aspects centralized in one single person. The demands for such a person are very high and, depending on the project, maybe even too high. Hedging one’s bets in the sense of utilizing more than one PO might, consequently, make sense as has already been indicated in literature [28]. We see this in all our cases. It is most visible in Foo’s case with three POs but Bar and Qux also distribute some aspects of what might be construed in strict Scrum doctrine (if there is such a thing) as the PO’s responsibilities. Hence, codifying strict roles might not always be advisable and a certain leeway might make sense. For example, appointing a senior member of a UX-design team to a part-time PO, assisting a continuous PO might be worth considering if actually employing two full-time POs with complimenting skill-sets is not viable due to project or financial constraints.

In regards to the very crucial aspect of filtering and interpreting user feedback, we also would like to point to the differentiation of roles, tools and channels as well as to a solid company culture as main factors for success. Furthermore, making conscious decisions about including qualitative as well as quantitative types of data in the development process seems highly advisable as well as economic. Regarding the operationalization of filtering and interpretative techniques: Quantitative filtering seems to be the more straightforward one – given thorough documentation in database form, user feedback can be quantified and analyzed rather easily. This data can supply very 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. This makes qualitative filtering a necessity. To use an analogy: In our interviews, we found certain similarities between this kind of filtering and qualitative sciences like ethnography: Deep immersion into a product’s user base and using the product oneself – getting a *feel* for it and forming *experience* – has been stated as very important and, again, 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 aspects 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 and interpretative action.

Thinking into the future, HCI and CSCW might provide help on the intersection of agile development and UCD in certain areas: As indicated by Qux' wishes for lean, almost Twitter-style feedback tools, Bar's utilization of a partly externalized user testing infrastructure or Foo's quick and easy in-house tool to work with app store reviews, properly (co-)designed tools to support agile and user centered processes are lacking. There are concepts from HCI and CSCW such as comprehensive situated user feedback and engagement mechanisms right inside of products, see e.g. [35] or leveraging modern mobile devices to facilitate relatively lean, event-contingent qualitative and quantitative data collection [7]. However, even if we as researchers might not necessarily like it, those concepts can sometimes be unwieldy and are not necessarily suitable for market-driven environments, necessitating collaborations between researchers and professionals[8]

6 Conclusion

We believe that there is no one-size-fits-all template for UCD/UUX and agile software development. The integration of user centered and agile principle is an artful business which necessitates many case-by-case decisions. However, we also believe that case studies such as the ones described in this contribution can help navigate at least parts of this difficulty – which incidentally is also we decided to keep our discussion on a relatively high level. Furthermore, we think that *some* principles might be abstracted and generalized to a certain extent. This is the intention behind the following bullet points – however, please keep in mind that they are grounded in three essentially qualitative case studies and can make no claim to completeness or applicability in all domains:

UCD is important for good UUX (and market success): This is the most obvious point and well-established in the scientific community but given the fact that multiple SMEs do not yet focus on UCD, it needs to be re-iterated.

Agile development is useful for user integration: Agile principles are well suited to be meshed with UCD methods and user integration. The combination also can have beneficial impact on company culture. However, there are still open questions (see below).

Role empowerment and company culture Multiple people with multiple perspectives need to have voices in the process and should to be able to

[8] An attempt at an explicitly simple and lean user feedback system similar to what Qux wished for is currently being developed open source led by our research group. It is called 'Shake' and interested parties are welcome to try it out and/or contribute on `http://github.com/UniSiegenCSCW/Shake`.

challenge decision making processes without retribution. Informal exchanges and grapevine is vital.

Multi-role / -channel / -tool is important A differentiated organizational structure is a solid base for UCD and agile development and should be constantly iterated upon. User integration at (too) isolated points might even be counter-productive. Triangulation is necessary.

Filtering and interpretation are necessary Not everything that a customer wants can be done or is actually a good idea and vice versa. Good ideas can be hard to come by. Qualitative and quantitative filtering mechanisms should be employed.

Filtering is not trivial: Staff needs to be educated, to actually use the product and to develop an appropriate frame of mind. Supportive ICT systems are useful but not necessarily available.

The PO is the critical point: A PO needs to make a significant amount of highly relevant decisions which is why she or he needs a grounded (multi-stage) base for those decisions and a quite comprehensive skill-set.

Consider hedging POs: It may be sensible to employ more than one PO or at least to treat the role more fluidly. If this is done, it is vital to establish and communicate the different responsibilities of the POs as not to impact the agile process negatively.

Acknowledgements

This contribution has been funded by the German BMWI (Federal Ministry for Economic Affairs and Energy) though the projects FKZ 01MU14001A and 01MU12026A.

References

1. Anderson, D.J., 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., Thomas, D.: Agile Manifesto (2001), http://agilemanifesto.org/
3. Beyer, H.: User-Centered Agile Methods (2010)
4. Bratteteig, T., Bjerknes, G.: User participation and democracy: A discussion of scandinavian research on system development. Scandinavian Journal of Information Systems 7(1) (Jan 1995), http://aisel.aisnet.org/sjis/vol7/iss1/1
5. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. 3, 77–101 (2006)
6. Chamberlain, S., Sharp, H., Maiden, N.: Towards a Framework for Integrating Agile Development and User-Centred Design. In: Extreme Programming and Agile Processes in Software Engineering, vol. 4044, pp. 143–153 (2006), http://dx.doi.org/10.1007/11774129_15$\delimiter"026E30F$npapers2://publication/doi/10.1007/11774129_15

7. Dax, J., Ludwig, T., Meurer, J., Pipek, V., Stein, M., Stevens, G.: Frames âĂŞ a framework for adaptable mobile event-contingent self-report studies. In: DÃŋaz, P., Pipek, V., Ardito, C., Jensen, C., Aedo, I., Boden, A. (eds.) End-User Development, Lecture Notes in Computer Science, vol. 9083, pp. 141–155. Springer International Publishing (2015), http://dx.doi.org/10.1007/978-3-319-18425-8_10
8. Draxler, S., Stickel, O., Winter, D., Stevens, G.: Nutzerintegration in Softwareprojekte durch Multi-Channel Feedback. In: Butz, A., Koch, M., Schlichter, J. (eds.) Mensch & Computer 2014 - Tagungsband. pp. 175–184. De Gruyter Oldenbourg, Berlin (2014)
9. Ehn, P., Kyng, M.: The collective resource approach to system design. In: Computers and democracy: A Scandinavian challenge, pp. 17–57. Avebury (1987)
10. Ferreira, J., Noble, J., Biddle, R.: Agile Development Iterations and UI Design. AGILE 2007 (AGILE 2007) (2007)
11. Floyd, C., Mehl, W.M., Reisin, F.M., Schmidt, G., Wolf, G.: Out of scandinavia: Alternative approaches to software design and system development. Human-Computer Interaction 4(4), 253–350 (1989), http://www.leaonline.com/doi/abs/10.1207/s15327051hci0404_1
12. Floyd, C., Reisin, F.M., Schmidt, G.: STEPS to software development with users. In: ESEC '89: Proceedings of the 2nd European Software Engineering Conference. pp. 48–64. Springer-Verlag, London, UK (1989)
13. Hansson, C., Dittrich, Y., Randall, D.: How to include users in the development of off-the-shelf software: A case for complementing participatory design with agile development. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006. HICSS '06. vol. 8, pp. 175c–175c (Jan 2006)
14. Hering, D., Kraft, X., Schwartz, T., Wulf, V.: Usability-Hindernisse bei Software entwickelnden KMU. In: Boll, S., Maaß, S., Malaka, R. (eds.) Mensch & Computer 2013 - Workshopband. pp. 9–18. Oldenbourg Verlag, München (2013)
15. von Hippel, E.v.: Democratizing Innovation. The MIT Press (Apr 2005), published: Hardcover
16. Holtzblatt, K., Beyer, H.: Making customer-centered design work for teams. Commun. ACM 36(10), 92âĂŞ103 (Oct 1993), http://doi.acm.org/10.1145/163430.164050
17. Isomursu, M., Sirotkin, A., Voltti, P., Halonen, M.: User Experience Design Goes Agile in Lean Transformation – A Case Study. 2012 Agile Conference pp. 1–10 (2012), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6298086
18. Keiningham, T.L., Cooil, B., Andreassen, T.W., Aksoy, L.: A Longitudinal Examination of Net Promoter and Firm Revenue Growth (2007)
19. Larusdottir, M., Cajander, A., Gulliksen, J., Cockton, G., Gregory, P., Salah, D.: On the integration of user centred design in agile development. In: Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational. pp. 817–820. NordiCHI '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2639189.2654836
20. Lee, J.C.: Embracing agile development of usable software systems. Conference on Human Factors in Computing Systems (2006), http://portal.acm.org/citation.cfm?id=1125451.1125784
21. Ley, B., Ogonowski, C., Mu, M., Hess, J., Race, N., Randall, D., Rouncefield, M., Wulf, V.: At Home with Users: A Comparative View of Living Labs. Interacting with Computers (2014), http://iwc.oxfordjournals.org/content/early/2014/07/29/iwc.iwu025.abstract

22. Lievesley, M.a., 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 p. 1025 (2006), http://dl.acm.org/citation.cfm?doid=1125451.1125647
23. Mao, J.Y., Vredenburg, K., Smith, P.W., Carey, T.: The state of user-centered design practice. Commun. ACM 48(3), 105–109 (Mar 2005), http://doi.acm.org/10.1145/1047671.1047677
24. Muller, M., Haslwanter, J., Dayton, T.: Participatory practices in the software lifecycle. pp. 256–297. Elsevier (1997), http://www.elsevier.com/wps/find/bookdescription.cws_home/524988/description#description
25. Reichheld, F.F.: The one number you need to grow. Harvard business review 81(12), 46–54, 124 (Dec 2003), http://www.ncbi.nlm.nih.gov/pubmed/14712543
26. Schwaber, K.: Scrum development process. In: Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA. pp. 117–134 (1995)
27. Silva, T., Silveira, M.S., Maurer, F., Hellmann, T., Paulo, U.D.S.a., Carlos, C.D.S.a., Carlos, S.a., Universidade, P.: User Experience Design and Agile Development : From Theory to Practice. Journal of Software Engineering and Applications 2012(October), 743–751 (2012)
28. Singh, M.: U-SCRUM: An Agile Methodology for Promoting Usability. Agile 2008 Conference (2008)
29. Steria, J.G.W., Alsos, O.: UX and Agile. Tutorial, NordiCHI conference 2010 (2010)
30. Strauss, A., Corbin, J.: Basics of Qualitative Research, vol. 3 (2008)
31. Sy, D.: Adapting Usability Investigations for Agile User-Centered Design. Journal of Usability Studies 2, 112–132 (2007), http://www.upassoc.org/upa_publications/jus/2007may/agile-ucd.pdf$\delimiter"026E30F$npapers2://publication/uuid/54FB82BC-3D3F-4B34-8E3C-7A02B631D618
32. Williams, L., Cockburn, A.: Agile software development: it's about feedback and change. Computer 36 (2003)
33. Wulf, V., Rohde, M.: Towards an integrated organization and technology development. In: Symposium on Designing Interactive Systems (DIS'95). pp. 55–64. ACM Press, Ann Arbor, MI, USA (1995), citeseer.ist.psu.edu/wulf95towards.html
34. Wulf, V., Rohde, M., Pipek, V., Stevens, G.: Engaging with Practices: Design Case Studies as a Research Framework in CSCW. In: Proceedings of the ACM conference on Computer supported cooperative work. pp. 505–512. ACM New York, Hangzhou, China (2011)
35. Yetim, F., Draxler, S., Stevens, G., Wulf, V.: Fostering continuous user participation by embedding a communication support tool in user interfaces. AIS Transactions on Human-Computer Interaction 4(2), 153–168 (Jun 2012), http://aisel.aisnet.org/thci/vol4/iss2/7