Margunn Aanestad
Tone Bratteteig (Eds.)

LNBIP 156

Nordic Contributions in IS Research

4th Scandinavian Conference on Information Systems, SCIS 2013
Oslo, Norway, August 2013
Proceedings

Springer

Lecture Notes
in Business Information Processing 156

Series Editors

Margunn Aanestad
Tone Bratteteig (Eds.)

Nordic Contributions in IS Research

4th Scandinavian Conference
on Information Systems, SCIS 2013
Oslo, Norway, August 11-14, 2013
Proceedings

Springer

Volume Editors

Margunn Aanestad
Tone Bratteteig
University of Oslo
Department of Informatics
Oslo, Norway
E-mail: margunn@ifi.uio.no
E-mail: tone@ifi.uio.no

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-39831-5 e-ISBN 978-3-642-39832-2
DOI 10.1007/978-3-642-39832-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013943825

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Digital Living

Our societies, organizations, and personal lives have been profoundly affected by information and communication technologies (ICTs). ICTs are embedded into everyday appliances, constitute core tools for organizational work, and are central mediators for our social life, both on the small, personal scale and on a political and societal scale. In this year's conference we invited participants to discuss experiences of "digital living" within a multitude of empirical settings. How are ICTs implicated in daily practices of care for elderly and frail people? How do organizations deal with their increasingly complex systems' portfolios? How can basic human capacities be utilized in interaction and collaboration with digital devices? How do people relate to each other when interaction is mediated through social networks? What are the societal effects of ICTs becoming ubiquitous in everyday situations, such as travel, shopping, everyday logistics, and community work?

The conference's keynotes were invited to present different takes on "digital living" and widen the scope of the discussion. Erik Fosse describes how medicine has been dramatically changed as a result of the introduction of digital technologies. Ulrike Schultze's research has addressed opportunities and challenges that digital technologies present for customer co-creation and peer-production. Specifically, she studies how "synthetic worlds," e.g., online games and virtual reality environments, function as media for organizational communication. Tom Igoe is known for his research on physical computing, and argues that when discussing the Internet of Things, we need to consider not just the things through which we communicate, but also the activities afforded by (and often impeded by) the capabilities of the networked ecology that we bring into being.

In response to the call for papers, distributed both on mailing lists and through personal networks, we received 18 submissions. Members of the Program Committee were invited to review the papers, and as a result of this review round, three papers were accepted directly. The authors of seven other papers were invited to revise and resubmit their papers according to the reviewers' comments. The resubmitted papers were reviewed by the editors, who decided to accept three more papers. The six accepted papers address "digital living" from different perspectives.

In the first paper, Netta Iivari addresses the role of usability specialists in commercial mobile application development. Specifically, she studies how the roles of both users and designers are changed when Open Source Software is being used. Software practices are also the topic of Lise Heeager and Peter Axel Nielsen's paper. They have studied learning among software developers who employ agile software development methods, and address knowledge transfer and

barriers for sharing knowledge between software teams within a company. In the third paper, Ellen Christiansen and Pernille Andersen question whether living with digital technologies implies "digital living." A study of use and non-use of digital home control systems led them to suggest a "design-with-users" discourse built on the collective resource approach. Alma Culén and Maja van der Velden also reflect on how to design with users, more specifically on how to practically include users who are easily left out of design work, like the elderly, children, and patients. In the fifth paper, Ivan Aaen examines the values that can facilitate and drive software innovation, i.e., software development that seeks to go beyond meeting requirements and aim to create novel, high-value solutions. In the sixth and final paper, Elena Parmiggiani and Marius Mikalsen offer a literature review of the emerging understandings of sociomateriality within information systems research.

We are grateful to all the authors who submitted their work to the SCIS 2013, and to all the Program Committee members who delivered detailed and constructive reviews. We hope you too will experience the resulting collection of papers as an interesting read.

June 2013

Margunn Aanestad
Tone Bratteteig

Organization

Conference Chair

Tone Bratteteig University of Oslo, Norway

Program Chair

Margunn Aanestad University of Oslo, Norway

SCIS 2013 Program Committee

Karin Axelsson, Sweden
Birgitta Bergvall-Kåreborn, Sweden
Claus Bossen, Denmark
Keld Bødker, Denmark
Alma Leora Culen, Norway
Gunnar Ellingsen, Norway
Sara Eriksen, Sweden
Miria Grisot, Norway
Erling Havn, Denmark
Karin Hedström, Sweden
Ola Henfridsson, Sweden
Stefan Henningson, Denmark
Jo Herstad, Norway
Jonny Holmström, Sweden
Arild Jansen, Norway
Pertti Järvinen, Finland
Netta Iivari, Finland
Karlheinz Kautz, Australia
Christina Keller, Sweden
Finn Kensing, Denmark
John Krogstie, Norway
Katarina Lindblad-Gidlund, Sweden
Rikard Lindgren, Sweden
Jan Ljungberg, Sweden
Carl Erik Moe, Norway
Eric Monteiro, Norway
Judith Molka-Danielsen, Norway
Bjørn Erik Munkvold, Norway
Christina Mörtberg, Sweden
Peter Axel Nielsen, Denmark
Jacob Nørbjerg, Denmark
Samuli Pekkola, Finland
Tero Päivärinta, Sweden
Pernille Bjørn Rasmussen, Denmark
Jeremy Rose, Denmark
Matti Rossi, Finland
Mikko Ruohonen, Finland
Ulrike Schultze, USA
Ada Scupola, Denmark
Kari Smolander, Finland
Reima Suomi, Finland
Lars Svensson, Sweden
Carsten Sørensen, UK
Virpi Tuunainen, Finland
Guri Verne, Norway
Pär Ågerfalk, Sweden

Table of Contents

‘Configuring the User and the Designer’ – A Critical Inquiry on Usability Work in the Company Open Source Software Development Context

Netta Iivari

Department of Information Processing Science,
University of Oulu, P.O. BOX 3000, 90014 Oulu, Finland
netta.iivari@oulu.fi

Abstract. This paper examines the complex interplay between Human Computer Interaction (HCI), open source software (OSS) and commercial mobile application development practices through a case study on usability work in company OSS development setting. The case company started using OSS in their products few years ago. There are usability specialists ‘representing the user’ in the development. The paper examines how the emergence of OSS affected the design process and outcome in the case. Specific focus will be on how the emergence of OSS contributed to the dynamics involved with ‘configuring the user’ and ‘configuring the designer’. The results show that usability specialists and developers collaboratively ‘configured the user’, but the emergence of OSS allowed users to participate earlier and have an increased prominence. Emphasis on users and usability as well as OSS ideology, on the other hand, in part ‘configured the designers’. However, non-computer-savvy users remained neglected.

Keywords: Open source software, human computer interaction, user-centered design, mobile application development.

1 Introduction

Information and communication technologies (ICT) have become an integral part of our everyday life. Web-based, mobile, ubiquitous, consumer and off-the-shelf ICT solutions have entered all spheres of our life, bearing important implications, among other issues, on the theory and practice of user participation, a decades old concern in Information Systems (IS) research. These recent developments have made it more difficult to work with users during development. Users may be scattered around the world or the user population may be even unknown until the product is in the market. One remedy has been to hire usability specialists to ‘represent the users’ in the development [10]. They are expected to ‘know the users’ and to ‘speak for the users’ during development [20]. This paper examines their work. On one hand, their work has been under empirical scrutiny for decades: empirical studies have been published already during 1980s. However, also recently a lot of studies have reported on usability work in practice in different kinds of development settings (see e.g. articles

M. Aanestad and T. Bratteteig (Eds.): SCIS 2013, LNBIP 156, pp. 1–17, 2013.

in [24, 25, 26]. New in this paper will be the emergence of open source software (OSS) in the commercial mobile application development setting where usability specialists are working, among other specialists.

This paper will examine the complex interplay between Human Computer Interaction (HCI), OSS and commercial mobile application development practices. The case company has traditionally had a strong usability focus and in mobile application development in general, the importance of users, usability and usability specialists have been acknowledged long ago [19, 27, 28, 29, 36]. On the other hand, the use of OSS is also becoming a common practice in commercial software development (see e.g. [13, 18]). OSS refers to software whose source code is 'available for anyone who wants to use or modify it' [35: 131], defined by its license. There is an increasing interest in OSS in the commercial setting, and it is expected that OSS will become highly influential in the future [13], due to which this study addresses a highly relevant development context, i.e. company OSS development context. There are, however, different ways companies can integrate OSS with their development. Companies may only use OSS as development tools or as part of their commercial products, but they can also release their source code for OSS communities to develop further. The OSS community can be an existing one or launched by the company. [13, 18, 35, 42].

The paper provides a critical examination of the process of 'configuring the user', i.e. of defining the identity of the future users and establishing the parameters for their future use practices [16, 41] during development. As in critical research the overall aim should always be the empowerment of the oppressed ones, this paper focuses on the design process from the viewpoint of the power-weak group of users. It has been warned that developers usually have the sole authority to 'configure the user' [1, 16], while HCI research strongly maintains that usability specialists should be allowed to do that [20, 45, 47, 44, 38] or at least to actively take part in that. The ability of usability specialists to make, or at least to influence, design decisions is a critical concern for HCI research, as it has been revealed that usability specialists have had difficulties in having any impact on the actual design solutions even though they were taking part in the development (cf. [2, 6, 14, 20, 33, 34, 40, 43]). Moreover, the emergence of OSS can be assumed to impact the process of 'configuring the user'. OSS brings along associated OSS communities who collaboratively create and further refine the solution. Therefore, it may not be only the usability specialists and developers inside companies anymore who are striving for the authority to 'configure the user'. In OSS development, the core developers usually have the authority to decide what to include in the code base [21], but when OSS enters commercial setting, OSS developers do not necessarily have that authority anymore. Altogether, OSS communities and companies are very different worlds due to which findings derived from OSS communities may not apply very well in companies. In commercial world there is a desire for profit maximization, while in OSS the emphasis is on 'collectivist, public-good community values', there being a tension between 'value for money' and 'acceptable community values' [13: 596] that may also have an effect on the design process and its outcome. Additionally, in OSS context the source code is available for everyone to use and modify, so one could assume that also users could

more actively take part in 'configuring the user' in the case of OSS [21, 23, 33, 34]. Finally, research has brought up that the 'configurers of the user' are also 'configured', by their organizations as well as by users, customers and networks extending outside their organizations. This paper will examine the dynamics involved with 'configuring the user' during the design process, while acknowledging also the dynamics involved with 'configuring the configurers of the user', i.e. the designers.

Some HCI studies have already reported on usability work in mobile application development [15, 19, 27, 28, 29, 36]. Also OSS development has gained increasing attention in the HCI community. Some studies already report on the benefits and challenges involved with integrating OSS development and usability work [2, 3, 4, 6, 8, 9, 23, 33, 34, 39, 40, 43]. Some studies have even touched upon the relationship between OSS development, commercial software development and usability work [2, 6, 9, 23, 31, 33], indicating that companies can use data gathered from OSS communities in their development and that companies may provide HCI resources for OSS communities. This paper will contribute to this line of research through a critical inquiry on the matter, asking "How does the emergence of OSS shape the dynamics involved with 'configuring the user and the designer' in the company OSS development context?" The existing research has not concentrated on the complex interplay between the divergent practices of HCI, OSS development and commercial mobile application development in any depth nor has it examined the actual influence of the emergence of OSS on the design process and outcome. In addition, there is a lack of critical research within the field of HCI [20], this paper trying to advocate critical technology research within HCI research.

The next section discusses the theoretical framework utilized in this paper. The third section reviews the existing research on usability work, including reviews in the mobile application and in the OSS development contexts. The fourth section presents the research method utilized, the case involved in this study and the procedures of data gathering and analysis. The fifth section presents the empirical results, the final sections discussing their implications.

2 Theoretical Lens

The paper relies on science and technology studies (STS) in the critical analysis of usability work in company OSS development context. Within this approach, it is assumed that technology developers always inscribe 'predictions about the world' into technological artifacts. Technologies are viewed to include scripts produced during development in which the envisioned users and use are realized. In these scripts, developers define projected, anticipated users with specific characteristics, competencies, motives, tastes and aspirations [1, 17, 41, 46]. Therefore, it is assumed that developers 'configure the user' during the development, i.e. they define and delineate the identity of the future users and establish parameters for their future use practices [16, 41]. In this paper the interest will be on who actually gets to act as the 'configurer of the user', the usability specialists and/or the developers as well as on how this gets accomplished in the new company OSS development setting. Who is allowed to act as the 'configurer of the user' in this setting?

Moreover, interestingly, it has been emphasized that not only are the users configured during development, but also the 'configurers of the user', i.e. the designers. Researchers have argued that the locale and processes of encoding need also to be considered, and the designers seen as configured – by users, by their own organizations' rules, regulations, methods and power structures as well as by broader actor networks extending outside their organizations [32]. The intra-organizational constraints that might be configuring the designers, i.e. shaping their identity and work practices, include company's methods and practices that need to be followed as well as the power relationships between different departments and between people inside particular departments. Users and customers may also have a lot of influence during the development: even though users in the research literature are typically viewed as the power-weak group, they are not necessarily such, but instead they might be quite influential in configuring the developers – as buyers, consumers or the ones making the final adoption decision. [32, see also 21] In this paper the focus will be on the dynamics involved with 'configuring of the users and the designers' in the company OSS development context, placing specific focus on how the emergence of OSS has shaped those.

3 Usability Work

The discipline of HCI, altogether, has been postulated as necessary as it 'represents the users' - an ignored group in systems design and computer science [10]. Initially, HCI research had strong background in the tradition of psychology. HCI work was laboratory-based experimental research, researchers focusing on general design principles contributing to user interface design and on evaluating existing systems. [5, 14] Users' role was consultative [11] at the most, i.e. they were allowed to comment on predefined design solutions in these evaluations, or usability specialists were 'representing the users' in their evaluations relying on general HCI principles and models without any contact with actual users. [5, 11, 14].

However, during 90s the focus shifted to the earlier phases of the development life cycle, i.e. to requirements specification and design that were claimed to be in need of usability specialists' contribution. It was claimed that it is not enough that users are represented during the last part of development, but instead they as well as usability specialists should earlier and more actively take part and have some decision-making power in the development [11, 20, 14]. At that time, the interest turned to anthropology and sociology as reference disciplines, and consequently the complexities of social, organizational, and cultural issues become more evident, and field studies and ethnographic inquiries in user population more popular, allowing users also informative role [11]; to act as providers of information in the development. [5, 10, 14] In addition, it was claimed that usability specialists should be given more authority to define the user group and their future use practices, and for these tasks they were provided with tools such as personas, scenarios, prototypes, and mock ups (e.g. [45, 47, 38]). Also users were to be invited into a more influential position in the design process through participatory design (e.g. [45, 38]).

Currently, usability work incorporates traces of all these historical developments. Usability evaluations of different kind are still popular, in laboratory settings as well as in the field (e.g. [44, 38]). Empirical inquiries in the potential or actual user population are also carried out during the requirements construction, not only during evaluation (e.g. [45, 44, 38]). In addition, the HCI literature recommends that usability specialists should be given authority to define the target user group and their future use practices based on their empirical inquiries and their state-of-the-art HCI knowledge [20, 14]. It is also acknowledged that users should be included as partners in the design process. However, this development has not been seen to remove the need of usability specialists' contribution and authority during development.

3.1 In Mobile Application Development

Research related to usability work has been carried out also specifically in the mobile application development context, in which it has been argued that mobility and tight development schedules pose challenges for usability work – the application will be used in mobile context with a mobile device with certain limitations and the development needs to be very fast in order to survive the harsh competition in the mobile application market (e.g. [15, 19, 27, 28, 29].

The studies have described what usability work entails in mobile application development. It seems that especially Contextual Design method [45] is widely used. The development has been started with contextual inquiries to understand users, their needs and the context of use [19, 29, 36]. Another widely used method seems to be paper prototyping, which is carried out together with users to improve the design solution during early phases of the development [19, 28, 29, 36]. Usability testing involving users as test participants in the later phase of the development is also mentioned in numerous studies [15, 19, 28, 29, 36].

Also new or modified methods have been suggested for mobile context: for example, diary and experience clip methods are mentioned as well-suited [27, 29]. As mobile applications are to be used in mobile context, they need to be evaluated in such a context as well [27]. Laboratory usability tests are not suitable for that reason, but instead users' experiences and feedback should be gathered in the mobile settings [15, 27, 29]. It is also recommended to tailor the methods used to fit the mobile context; for example contextual inquiries are recommended to be used in a light weight way to gain insights but not spending too much time or resources [19, 29, 36].

3.2 In OSS Development

Some research touching upon usability work in OSS development also exists. As the source code of OSS must be available for anyone to use and modify, everyone needs to be considered as potential user and developer of OSS. Users can freely contribute to OSS development through bug trackers, discussion forums, mailing lists and such. Traditionally, OSS users have been developer-users, technically competent people capable to contribute to the development through coding. Nowadays, nevertheless, a large proportion of OSS users may not be capable or willing to contribute to development by utilizing the above mentioned means. [6, 9, 33, 34, 39].

To take care of the needs of these less-technically-competent' uses, some usability methods have already been recommended for OSS development. Empirical usability testing has been suggested [2, 33, 34, 39, 43]. It is suitable as there might be a large user base in OSS projects reachable for testing purposes [43]. On the other hand, usability inspections based on general HCI principles have also been brought up [43], as well as usability discussion lists [8, 33] and design areas for brainstorming and discussing the evidence from user studies and design guidelines [4, 40].

The literature also brings up the importance of usability specialists' participation in OSS development [2, 4, 6, 9, 23, 33, 39, 40, 43]. They should define user requirements and user profiles [9], analyze and condense user contributions in the discussion forums as well as engage in design discussions in OSS projects [33]. However, it has been noted that usability work might be challenging in OSS projects, as there might be very few usability specialists and those few might be isolated, alienated and without any decision-making power [4, 6, 8, 9, 34, 40].

It has also been brought up that the companies involved in OSS development could provide professional usability specialists and HCI guidelines for OSS projects [2, 6, 22, 23, 34]. The usability specialists hired by companies could take the responsibility to carry out usability tests and user studies [23, 34]. However, it has been warned that there might be difficulties in combining corporate and heavyweight usability processes with 'open source culture and ideology' [6, 8, 9, 33, 43], indicating that these two practices – OSS and HCI development– do not necessarily very easily mix.

4 Research Design

The paper reports results from an interpretive inquiry within which the research concepts and problem statements should be local and emergent rather than elite/a priori ones [12] and researchers should attempt to understand and make sense of the world, not to explain in the predictive sense [30: 69]. Research subjects are to be considered as active meaning makers just like the researchers, who should collaboratively work out the key understandings of the case [12, 30]. The paper examines a case that is a software development unit of a large, global corporation. The unit is responsible for the OSS development in the company. The unit has also a strong background in usability work: there are usability specialists hired to 'represent the users' (cf. [10]) in the development, and the number of them has even increased during recent years as the importance of satisfied users and high level usability have become more important. The unit mainly uses OSS as part of their commercial products; OSS is used as a basis on top of which applications are developed, the user interface remaining closed source. In some cases the source code of the whole product has been released. The user interface code is kept closed source as user interface solutions are perceived to provide the competitive edge and a legal check would otherwise be needed that is perceived complicated and expensive.

This case was selected; as it offers a rich setting to analyze the interplay between commercial mobile application, OSS and HCI development practices. There are usability specialists and developers taking part in the design process and it is interesting to inquiry whether the emergence of OSS has changed the dynamics

involved with it. Access to the unit was gained through negotiating it with the unit managers. Different kinds of empirical data have been gathered from the unit. This paper relies specifically on data gathered by interviewing the personnel of the unit. The already finished company OSS projects had consisted of developers, usability specialists, testers and project managers. A unit manager searched for suitable interviewees, and found two knowledgeable developers, two knowledgeable usability specialists and one knowledgeable project manager to be interviewed. In the interviews, characteristics of the company's OSS projects were discussed as well as issues related to OSS development inside the company and usability work in the projects. Additional material contains field notes from three meetings organized with the unit management. The notes include same issues as those discussed in the interviews but on a more general level and in a less systematic manner. Moreover, blogs the personnel of the unit had kept related to OSS development and usability work and some OSS communities' websites were included as research material.

All the material was transformed into written form; the interviews were transcribed. Afterwards, the material was analyzed by focusing on the following aspects: 1) how did the emergence of OSS shape the dynamics involved with 'configuring the user'; and 2) how did the emergence of OSS shape the dynamics involved with 'configuring of the designer'.

5 Empirical Insights

5.1 'Configuring the User'

In the beginning of the OSS projects inquired in this study the usability specialists were in a relatively influential position. The projects started by settling the user group for which the solution was to be developed. This was done by the usability specialists: "*The first (phase) (…) tries to understand the people involved, the customers' needs and clients' needs. (…) We try to do some benchmarks and a persona, at least a simple persona, just to keep track of everything; we don't go too much deep in personas."* (Usability specialist) However, actual users were not contacted, but instead the usability specialists relied on their own expertise: "*This was, let's say technically quite challenging case and a challenging one from the viewpoint of user. It requires certain kind of interest in technical issues to start to use it. So, we did not carry out a traditional user study in the beginning to figure out the characteristics of the end user. We had to assume that they are technically quite capable.*" (Usability specialist) "*Well, at first we sat down and thought that we are now making this kind of (an application) and thought that the display is this kind of (a display). (…) So, this is the starting point and otherwise we have a lot of freedom to do what we want.*" (Manager) It was assumed that the target user group was "*super geeks*" but the goal was to move towards "*mass market"* (Field notes). Overall, the personnel inside the development organization imagined and afterwards briefly described what kind of users (i.e. '*technically capable*, *super geeks*') the target users would be.

After settling the target user group, design solutions were iteratively produced by the usability specialists. The work started by defining the users' future use practices, after which user interface solutions were designed. "*We started by building use cases in a user centered way. We described the normal use situations that can be related to the software like this. After that we moved to producing the rough user interface.*" (Usability specialist) There was active collaboration among the usability specialists and developers during this phase: "*We sat there together for an hour and brainstormed and then they (the usability specialists) left and came back within a couple of days and asked that would it be like this?*" (Manager) "*Always, when we were able to produce something finished by our opinion, we went to present it to the developers and asked that is this possible?*" (Usability specialist) The usability specialists had the authority to produce the initial designs, however, the developers having the right to check them afterwards. The collaboration was characterized as unofficial and active: "*There were fewer meetings. It was with the implementation team like we just went there and asked. We did not produce a list to be discussed in a meeting. It was like active communication all the time.*" (Usability specialist).

Evaluations from the users' point of view were also carried out. Both expert evaluations relying on the HCI knowledge of the usability specialists and empirical user testing with actual or potential users were carried out: "*First we produce a rough user interface (…) and afterwards (…) we carry out expert evaluation. In them we use one or more usability specialists and modify the user interface according to them, and then move it to a more detailed level. After that we make a simulation of the user interface and carry out a traditional usability test in a laboratory with a sufficient amount of users, from six to ten per iteration.*" (Usability specialist) During empirical evaluations also the OSS communities entered the scene. They were utilized as providers of user feedback and design ideas in a fast and easy way: "*These open source software, they create a lot of communities, so it's a really, really rich place to gather user feedback. (…) What we try to do is to use open source users, even the developers, as partners in the beginning and in the design phase.*" (Usability specialists) "*Actually what we did was: hi, let's do a prototype in, for example, couple of weeks and see how it goes. The user interface is very, very, very simple. What we did was to make it and release it. (…) Involvement was all through emailing community: what do you like to see, what do you like to have? (…) We did use the community to find out more and to find out improvements.*" (Usability specialist) OSS users were contacted through mailing lists and discussion forums: "*We gained comments of it, when we read the Internet forums afterwards."* (Usability specialist) Through these means, users were mainly allowed to provide feedback to the predefined design solutions. However, in cases the source code had been released, users had also contributed code to the application: "*We gave all the code to the community and gained their acceptance. (The application) has been made expandable. Users can make plug-ins. The infrastructure has been built.*" (Developer) Usability work was argued to benefit from the emergence of OSS in commercial setting – user involvement became faster and cheaper: *"In the open source project*

you actually get the final users much faster for you from all around the world. For usability perspective, I would say that it's simpler and cheaper to do in the open source." (Usability specialist).

5.2 'Configuring the Designer'

A manager identified two types of developers in the company: "*There is a clear differences between an OSS coder and (a closed source operating system) coder. The OSS coders are excellent.*" (Field notes) For some developers OSS "*is just code that is available there (laughing)*" (Manager). Therefore, not all developers were influenced by the emergence of OSS in the company. Some, on the other hand, embraced OSS development and defined it as "*some sort of charity work*" and separated "*actually open source*" from the work they did in the company. Related to that, they said "*of course you can find all these firms over the sea, whose names I'm not going to mention, who have their offices in Redmond (laughing), who try to be supporters of OSS. It makes me laugh to a certain extent.*" (Developer) Among the OSS savvy employees, learning and knowledge sharing were emphasized as characteristics of OSS development: "*One great thing is that when you are using open source, you always learn. I can see other people code. I can learn from their ideas, if there is expertise with the given language and so on.*" (Developer) "*I would describe it as learning and collaboration. (….) It's a place where you can learn: (…) It's like a social network of knowledge. It's quite nice how it goes. Quite nice that you can actually be part of it, teaching a little bit and learning a lot more.*" (Usability specialist) Therefore, there were employees working inside the company who emphasized that even though they worked in the company, they were 'true OSS developers or advocates' by heart, adhering to the OSS ideology that argues for free software, free information, sharing, gift giving and altogether 'collectivist, public-good community values' (e.g. [7, 13, 37, 42]).

This part of the personnel preferred releasing the source code. For instance in a blog post after a release of the source code of an application, a usability specialist rejoiced that "*We managed to do it! After this huge amount of work that has been mainly connected with persuading people to believe that it is important to open source it, now the source code of (the application) is released with (a OSS license).*" (Usability specialist) This post was commented by a user announcing that now; finally, he could start fixing its bugs. Related to a connected blog post, there also were numerous comments congratulating the development team for releasing the source code, i.e. succeeding to '*free (the application)*', the management being blamed for the delay in doing so. However, not always was the source code released and it was acknowledged that this might cause anger for some OSS users, but it was not seen as an issue for those appreciating high quality design solutions: "*If the user interface is closed, of course it generates a little bit of anger in some people, but I think sometimes they are also ok," if you guys do a nice job, we don't care as much as the features are being implemented."*(Usability specialist).

It was perceived that the emergence of OSS had made the development faster, allowed more freedom in the projects and enabled putting even more time and effort on producing quality design. The development was faster as: "*I do not need to create everything from scratch. So this is why I can concentrate on creating new things, I don't need to bother about the old things that everybody knows how to do. (…) If we created something that was closed, traditional process, we could never have achieved, probably it would have taken too much time to achieve what was achieved.*" (Developer) The personnel also perceived that they had more freedom to do their work in OSS projects compared to their traditional projects: "*We have had pretty much freedom to do the user interface. It is a pretty rare situation (in the company). (…) We carried out usability work and user interface design process the normal way, there was no difference, but the restrictions and starting point for the users interface, there was freedom.*" (Usability specialist) The company had allowed the developers to adopt some aspects of the open OSS development model: some OSS tools (Bugzilla) and practices (distributed code review): "*There is a small community inside the company, who accept code to the public main tree. If it is not good, it is discussed. Many people review the same code. (…) The aim is not to spend a lot of time on that. (…) The review concerns other projects' code, not own. Outsiders tell whether there is something that needs to be fixed; it is not own code. Then there is also feedback from the community*" (Developer) Finally, the use of the OSS had given more time for producing quality designs for users. The personnel could evaluate the finished solution earlier, to spend more time on design and to make late changes based on the feedback: "*We were able during much earlier phase to handle the finished software. (…) Usually it takes pretty long time to make a simulation of it. (…) Here we were able to experiment in practice what it looks like and how it feels like in real use. (…) This way it affected, positively."* (Usability specialist) *"The ability of doing late changes that of course the open source project provided us, because they give us ground work that is solid and mature. Then we can just build the experience on top. So actually in the product competition, we have time to do more testing or more designing and more late changes.*" (Usability specialist).

6 Discussion

This paper critically examined usability work in company OSS development context by utilizing a STS oriented framework. Table 1 summarizes the empirical results revealing how the emergence of OSS in the company context had influenced the dynamics involved with 'configuring the user' as well as with 'configuring the configurers of the user', i.e. the designers.

Table 1. Configuring the User and the Designer in the Company OSS Development Context

	Empirical Insights
Configuring the User and the Designer	- Usability and users perceived as important in the company, giving authority to usability specialists, making developers sympathetic to users and usability - Usability specialists hired to 'represent the users': defining users in personas and their work practices in use cases based on their HCI and/or domain knowledge and imagination, and iterate them with mock ups - Developers iterate the designs with usability specialists and implement them afterwards - Usability specialists comment on the designs in expert evaluations relying on their HCI expertise - Users comment on the designs in usability testing and OSS forums - Users contribute through coding in the OSS environment **Influence of OSS on 'Configuring the User'** - Emphasis on freedom for users to contribute: OSS users comment on the 'configurations of the user' in OSS forums and contribute to the 'configuration of the user' through coding - Emphasis on 'configuring the user': more time spent on early and fast evaluation as well as design and iteration **Influence of OSS on 'Configuring the Designer'** - Emphasis on freedom for users to contribute: OSS oriented designers demand releasing the source code so that OSS users can contribute - Emphasis on freedom for designers to contribute: less restrictions placed on 'configuring the user' by the case company - Emphasis on 'configuring the user': more time spent on early and fast evaluation as well as design and iteration - Emphasis on open and participatory development: more extensive user and designer collaboration, while no power over designers by users

6.1 Critical Remarks on the Practice of Usability Work

This paper offered a critical account of usability work in company OSS development context by utilizing a STS oriented framework, filling in the gap of critical HCI research by examining from the viewpoint of the power-weak group of users how the design gets done. The paper examined the dynamics involved with 'configuring the user' in the development, acknowledging that also the configurers of the user get configured. Existing STS as well as HCI research has warned us that users are the power-weak group in technology development, developers having the authority to 'configure the user' [10, 16, 47]. Usability specialists have been positioned as needed for representing the ignored users [10], but usability specialists have encountered difficulties in their work having any impact on the actual design solutions [2, 6, 14, 20, 33, 34, 40, 43]. Regarding the case analyzed in this paper, there were usability specialists hired to 'represent the users' in the development, which is quite typical in companies operating in the product development context [20, 14]. In this case as well as typically in product development organizations, usability specialists were allowed a consultative role, i.e. they provided feedback to the already made design decisions through expert evaluations and empirical usability testing (cf. [20, 14]). Interestingly, the usability specialists were also allowed a participative role in the development

(cf. [20]), i.e. they were allowed to define the future users and their future use practices, the developers then commenting on them and implementing the solutions afterwards. In these instances the usability specialists clearly had authority to 'configure the user', which is not always the case in commercial software development setting [20, 14].

Users were also allowed the consultative role to comment on these predefined design solutions. Typically this takes place in empirical usability evaluation sessions, which was the case also in this case. In this case, moreover, OSS users had entered the scene by utilizing OSS means available, i.e. mailing lists and discussion forums (cf. [23, 31]). These users, assumingly the "*super geeks*" and "*nerds"*, were also allowed the consultative role, i.e. to provide feedback to the existing design decisions, and occasionally even a participative or designer role, i.e. to take active part in producing the solutions, therefore having increased ability to influence the 'configuration of the user'. In this case, furthermore, it was highlighted that due to the use of OSS, users were able to take part earlier and more easily through OSS forums, and even late changes based on user feedback were possible. However, through these means one usually contacts rather computer-savvy ("*super geek, nerd"*) users. As mentioned, in these projects the target user group was defined to consist of this type of users, due to which it is not such a problem. Nevertheless, for the non-computer-savvy user population, there were no efforts to allow them any ability to influence the outcome of the design process – even though future plan was to reach the '*mass market*'. Reliance on OSS users to give feedback and to contribute code might create an illusion of active user participation; however, OSS users may resemble developers too much. Cooper [47] reminds us that developers represent a particular kind of species, homo logicus, who is nevertheless designing for homo sapiens. Thus, it is important to be in contact with homo sapiens during the development, i.e. to prevent the developers designing the solutions for them and for the like-minded. In OSS communities there is a great risk of finding mainly homo logicus people to comment and to contribute. All in all, one can say that in connection to those solutions that were released as OSS, the computer-savvy users were empowered to modify and further develop the solutions although without any actual influence on the solution developed by the designers. Furthermore, not all solutions were released as OSS and the less-technically-competent users were neglected, altogether.

Moreover, not only are the users 'configured' during development, but also the designers, which might have also some implications on the design process and outcome. In this case, high quality design for users seemed to be in rather important position and everyone seemed to accept this. Therefore, users and usability can be argued to define the designers' identity and work practices to a certain extent. Hence, usability work was carried out in the projects and nobody seemed to resist that. OSS can also be argued of having configured some of the designers in this case. There were developers labeled as 'talented OSS developers' and some usability specialists also positioned themselves as advocates of OSS. They wished to emphasize that they were 'true' OSS developers or advocates by heart, knowing the limitations of company OSS development and associated rhetoric. They celebrated the release of their source code and therefore emphasized the freedom for users to contribute to development. They relied on open and participatory development model and

experienced also themselves as having more freedom to do their job as well as more time for 'configuring the user' compared to company's other projects. However, users were not positioned as decision-makers in the design process, but only as OSS expanders after the release. The case arouses the question of to what extent these OSS oriented employees feel comfortable to strive for the company and management goals, while explicitly emphasizing the basic OSS values such as free software, free information and gift giving [7, 37, 42]. It would not be surprising to find out that these kinds of employees experience some difficulties in coping with the tension between 'value for money' and 'acceptable community values' [13: 596, see also 42].

6.2 Practical Implications

There are also some practical implications for usability work identifiable from this study. The emergence of OSS had resulted in many positive aspects from the viewpoint of commercial mobile application development and usability work. It had resulted in the projects having more time for innovating new things and for designing quality solutions for users. More time could be spent on design and iteration, as OSS solutions speeded up the development work. The emergence of OSS, however, seemed mainly to contribute to the evaluations that were carried out in a faster and easier manner when using OSS solutions. In addition, user feedback was gathered even earlier than had been the case. These issues, on the other hand, are very appealing for the mobile application development context, in which tight schedules might otherwise inhibit usability work [19, 28, 29, 36].

The OSS forums made it easy and fast to contact users in their natural (mobile) settings. These forums, altogether, can be used for alleviating the problem of the lack of time and resource in fast-paced mobile application development as well as the problem of gathering user feedback to mobile applications in laboratory settings instead of in mobile field settings [15, 23, 27, 29]. Of course, OSS forums are not available in every relevant mobile setting. If they nevertheless are available (e.g. in mobile devices having internet access), their use context surely resembles more natural mobile context than a usability laboratory. Therefore, OSS forums might be highly useful in evaluating mobile applications. Through them, users can spontaneously provide user feedback in their actual context of use.

A positive issue in the case from the viewpoint of HCI research is also that the developers seemed to appreciate usability and users a lot, which is not necessarily the case in OSS projects (cf. [2, 6, 33, 34, 40, 43] nor in commercial software development (cf. e.g. [20, 24, 14]). The use of OSS had contributed to usability work by allowing more time to be spent on it. This study, among others, indicates that companies indeed can provide useful resources and know-how to OSS communities (cf. [2, 6, 22, 34]). A problem is that the contribution of the usability specialists might affect only that part of the solution that is kept closed source. In this situation the OSS communities naturally cannot benefit from the usability specialists' work.

However, the emergence of OSS did not seem to help much in enabling users' to take part in the requirements construction or in the design phase (see also [23]). Problematic from the viewpoint of usability work is that in the beginning there was no

contact to any user group, but the usability specialists instead simply imagined and afterwards briefly described the intended users. In the projects of the case unit the target user group was defined to be technically-savvy '*nerds*' and this kind of 'developer-user-pool' available in the OSS communities was utilized during evaluations and some this kind of users had even contributed code, but not even this kind of users were involved during requirements construction and design. Moreover, the less-technically-competent users were neglected, altogether.

7 Conclusions

This paper offered a critical examination of the practice of usability work in company OSS development context by utilizing a STS oriented framework. The analysis sensitized us to the constant construction of user and designer (usability specialist and developer) identities and work practices that inevitably takes place during software development. The emergence of OSS influenced the way how the design process took place. Especially positive consequences were identified. The results showed that the usability specialists and developers hired by the company collaboratively settled the user group and designed users' future use practices. The use of OSS allowed users to participate earlier and have an increased emphasis. The OSS users were relied on during evaluations and in some cases even allowed to contribute code. OSS also in part defined the identity and work practices of these designers, by allowing them an identity of an OSS developer or advocate, by placing emphasis on the freedom for users to contribute and by enabling these designers also to have more freedom to do their job, to utilize a more open and participatory development model and to put more time and effort to design and iteration. However, it was warned that the involvement of OSS users might give the designers an illusion that there is extensive user participation taking place, although in OSS communities the user population tends to be quite technologically-savvy and for that reason resemble more the developers' mindset than that of the users'. An interesting path for future work would be to analyze how the above-discussed group of 'OSS developers' and like-minded usability specialists cope inside the companies: are there value conflicts and conflicts of interest that they feel disturbing (cf. [42])? Interesting would also be to include more empirical cases into the analysis to see whether the aspects identified in this study appear also in other development settings.

Acknowledgements. This research has been funded by the Academy of Finland.

References

1. Akrich, M.: The De-Scription of Technical Objects. In: Bijker, W., Law, J. (eds.) Shaping Technology/Building Society. Studies in Sociotechnical Change, pp. 205–224. MIT Press, Cambridge (1992)
2. Akrich, M., Latour, B.: A Summary of a Convenient Vocabulary for the Semiotics of Human and Nonhuman Assemblies. In: Bijker, W.E., Law, J. (eds.), pp. 259–264 (1992)

3. Andreasen, M., Nielsen, H., Schrøder, S., Stage, J.: Usability in Open Source Software Development: Opinions and Practice. Information Technology and Control 25(3A), 303–312 (2006)
4. Bach, P., Carroll, J.: FLOSS UX Design: An Analysis of User Experience Design in Firefox and OpenOffice.org. In: Proc. 5th International Conference on Open Source Systems, Skövde, Sweden, June 3-6, pp. 237–250 (2009)
5. Bach, P., DeLine, R., Carroll, J.: Designers Wanted: Participation and the User Experience in Open Source Software Development. In: Proc. Conference on Human Factors in Computing Systems, pp. 985–994. ACM Press, New York (2009)
6. Bannon, L.: From human factors to human actors: The role of psychology and human-computer interaction studies in system design. In: Greenbaum, J., Kyng, M. (eds.) Design at Work: Cooperative Design of Computer Systems, pp. 25–44. Lawrence Erlbaum, Hillsdale (1991)
7. Benson, C., Müller-Prove, M., Mzourek, J.: Professional usability in open source projects: GNOME, OpenOffice.org, NetBeans. In: Extended Abstracts of the Conference on Human Factors in Computer Systems, pp. 1083–1084. ACM Press, New York (2004)
8. Bergquist, M., Ljunberg, J.: The power of gifts: organizing social relationships in open source communities. Information Systems Journal 11, 305–320 (2001)
9. Beyer, H., Holtzblatt, K.: Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann Publishers Inc., San Francisco (1998)
10. Bødker, M., Nielsen, L., Orngreen, R.N.: Enabling User Centered Design Processes in Open Source Communities. In: Aykin, N. (ed.) Usability and Internationalization, Part I, HCII 2007. LNCS, vol. 4559, pp. 10–18. Springer, Heidelberg (2007)
11. Çetin, G., Verzulli, D., Frings, S.: An Analysis of Involvement of HCI Experts in Distributed Software Development: Practical Issues. In: Schuler, D. (ed.) HCII and OCSC 2007. LNCS, vol. 4564, pp. 32–40. Springer, Heidelberg (2007)
12. Cooper, A.: The Inmates Are Running the Asylum: Why high-tech products drive us crazy and how to restore sanity. Macmillan, Indianapolis (1999)
13. Cooper, C., Bowers, J.: Representing the users: Notes on the disciplinary rhetoric of human-computer interaction. In: Thomas, P. (ed.) The Social and Interactional Dimensions of Human-Computer Interfaces, pp. 48–66. Cambridge University Press, Cambridge (1995)
14. Damodaran, L.: User involvement in the systems design process – a practical guide for users. Behaviour & Information Technology 15(16), 363–377 (1996)
15. Deetz, S.: Describing Differences in Approaches to Organization Science: Rethinking Burrell and Morgan and Their Legacy. Organization Science 7(2), 191–207 (1996)
16. Fitzgerald, B.: The Transformation of Open Source Software. MIS Quarterly 30(3), 587–598 (2006)
17. Gallant, L.: Ethnography of communication approach to mobile product testing. Personal and Ubiquitous Computing 10, 325–332 (2006)
18. Grint, K., Woolgar, S.: The Machine at Work. Technology, Work and Organization. Polity Press, Cambridge (1997)
19. Heikinheimo, H., Kuusisto, T.: The use of embedded open source software in commercial products. In: Proc. of the 13th European Conference on Information Systems, Turku, Finland, June 14-16 (2004)
20. Holtzblatt, K.: Customer-centered design for mobile application. Personal and Ubiquitous Computing 9, 227–237 (2005)

21. Iivari, N.: Understanding the Work of an HCI Practitioner. In: Morch, A., Morgan, K., Bratteig, T., Ghosh, G., Svanaes, D. (eds.) Proc. 4th Nordic Conference on Human Computer Interaction, Oslo, Norway, October 14-16, pp. 185–194 (2006)
22. Iivari, N.: Empowering the Users? A Critical Textual Analysis of the Role of Users in Open Source Software Development AI & Society 23(4), 511–528 (2009)
23. Iivari, N.: Culturally Compatible Usability Work - An Interpretive Case Study on the Relationship between Usability Work and Its Cultural Context in Software Product Development Organizations. Journal of Organizational and End User Computing 22(3), 40–65 (2010)
24. Iivari, N.: Participatory Design in OSS Development: Interpretive Case Studies in Company and Community OSS Development Contexts. Behaviour & Information Technology 30(3), 309–323 (2011)
25. Iivari, N., Hedberg, H., Kirves, T.: Usability in Company Open Source Software Context. Initial Findings from an Empirical Case Study. In: Proc. 4th International Conference on Open Source Systems, Milan, Italy, September 7-10 (2008)
26. Interacting with Computer 18(4)
27. International Journal of Human-Computer Interaction 21(2)
28. International Journal of Technology and Human Interaction 5(3)
29. Isomursu, M., Kuutti, K., Väinämö, S.: Experience Clip: Method for User Participation and Evaluation of Mobile Concepts. In: Proc. Participatory Design Conference, Toronto, Canada, July 27-31, pp. 83–92 (2004)
30. Jokela, T., Koivumaa, J., Pirkola, J., Salminen, P., Kantola, N.: Methods for quantitative usability requirements: a case study on the development of the user interface of a mobile phone. Personal and Ubiquitous Computing 10, 345–355 (2006)
31. Kangas, E., Kinnunen, T.: Applying User-Centered Design to Mobile Application Development. Communications of the ACM 48(7), 55–59 (2005)
32. Klein, H., Myers, M.: A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly 23(1), 67–94 (1999)
33. Latour, B.: Where Are the Missing Masses? The Sociology of a Few Mundane Artifacts. In: Bijker, W., Law, J. (eds.) Shaping Technology/Building Society. Studies in Sociotechnical Change, pp. 225–258. MIT Press, Cambridge (1992)
34. Lewis, S.: Using Online Communities to Drive Commercial Product Development. In: Extended Abstracts on the Conference on Human Factors in Computing Systems, pp. 2039–2044. ACM Press, New York (2008)
35. Mackay, H., Carne, C., Beynon-Davies, P., Tudhope, D.: Reconfiguring the User: Using Rapid Application Development. Social Studies of Science 30(5), 737–757 (2000)
36. Mayhew, D.: The Usability Engineering Lifecycle: A practitioner's handbook for user interface design. Morgan Kaufmann Publishers Inc., San Francisco (1999)
37. Nichols, D., Twidale, M.: The Usability of Open Source Software. First Monday 8(1), 21 p. (2003)
38. Nichols, D., Twidale, M.: Usability Processes in Open Source Projects. Software Process Improvement and Practice 11, 149–162 (2006)
39. Niederman, F., Davis, A., Greiner, M., Wynn, D., York, P.: A Research Agenda for Studying Open Source I: A Multilevel Framework. Communication of the Association for Information Systems 18, 129–149 (2006)
40. Olsson, T., Lehtonen, M., Pavel, D., Väänänen-Vainio-Mattila, K.: User-Centered Design of a Mobile Application for Sharing Life Memories. In: The Proc. of Mobility, Singapore, September 10-12, pp. 524–531 (2007)

41. Rolandsson, B., Bergquist, M., Ljungberg, J.: Open source in the firm: Opening up professional practices of software development. Research Policy 40(4), 576–587 (2011)
42. Rosson, M., Carroll, J.: Usability Engineering: Scenario-based Development of Human-Computer Interaction. Morgan-Kaufman, San Francisco (2002)
43. Stewart, K., Gosain, S.: The Impact of Ideology on Effectiveness in Open Source Software Development Teams. MIS Quarterly 30(2), 291–314 (2006)
44. Terry, M., Kay, M., Lafreniere, B.: Perceptions and Practices of Usability in the Free/Open Source Software (FOSS) Community. In: Proc. Conference on Human Factors in Computing Systems, pp. 999–1008. ACM Press, New York (2010)
45. Twidale, M., Nichols, D.: Exploring Usability Discussions in Open Source Development. In: Proc. 38th Hawaii International Conference on System Sciences, HICSS. IEEE Computer Society Press (2005)
46. Woolgar, S.: Configuring the User: the case of usability trials. In: Law, J. (ed.) A Sociology of Monsters. Sociological Review Monograph, vol. 38, pp. 57–102. Routledge, London (1991)
47. Zhao, L., Deek, F.: Improving Open Source Software Usability. In: Proc. of the 11th Americas Conference on Information Systems, Omaha, USA, August 11-14, pp. 923–928 (2005)

Agile Software Development and the Barriers to Transfer of Knowledge: An Interpretive Case Study

Lise Heeager and Peter Axel Nielsen

Socio-Interactive Research Centre, Department of Computer Science
Aalborg University, Denmark
{liseh,pan}@cs.aau.dk

Abstract. Agile practices to systems development are believed to depend largely on the developers' competences, experience and knowledge and to a lesser degree on formal development processes and methods. In this paper we investigate the knowledge transfer and barriers to the transfer of agile development practices in an interpretive case study. The case company is a pharmaceutical firm where we studied how they develop software and how they transfer their own experience. Based on the literature we develop an initial framework of barriers to knowledge transfer and apply it to interpret the case study. From this case study we are able to discuss the initial framework and extend it to a framework of knowledge transfer of agile practices. The framework provides a better understanding of the barriers to knowledge transfer of agile practices. The paper contributes with (1) an application of the framework to explain knowledge transfer and barriers, and (2) specifically explicate potential barriers hindering knowledge transfer of agile practices. This has implications for the implementation of agile development practices.

Keywords: knowledge transfer, agile software development, barriers.

1 Introduction

Information systems development has for more than a decade been concerned with a change of approaches and perspectives towards more the more agile [1], [2] and the agile ideas, methods, and practices [3], [4]. The adoption of agile ideas by one team in a software company is only a first step in a more widespread diffusion and adoption where many teams in a company and even whole companies are applying agile approaches. In this paper we focus on the transfer of knowledge and experience with agile practice from one team in a company to another team in the same company.

Inside organizations reside much unknown and untapped knowledge that can be helpful for the whole organization [5]. Identifying and transferring knowledge across boundaries in an organisation and between teams is a powerful mechanism for improving the productivity of an organization and create a significant competitive advantage [6]. The transfer of knowledge within organizations is however often laborious, time consuming and difficult [7], [8], [9]. In software development transferring experience from one project to another is especially difficult [10].

M. Aanestad and T. Bratteteig (Eds.): SCIS 2013, LNBIP 156, pp. 18–39, 2013.

A software organization's main asset is its intellectual capital [11] and software development is highly tacit with knowledge rooted in actions [12], [13], [14], and with developers working under time pressure, specialized requirements as well as different cultures from the users. This is particularly the case for agile software development [15], [16], [17]. Few studies report on practical knowledge of agile approaches being transferred, e.g., [18] . It is clear that while knowledge transfer is desirable in agile software development it also meets barriers – some which are easily overcome and some which are inherently difficult [13], [19], [20]. Existing research, however, address knowledge transfer within agile software teams and not between teams.

In this paper we therefore address how knowledge on agile practices is transferred from one development team to another and in particular the barriers to such knowledge transfer. We therefore pose the research question: Which barriers can prevent the transfer of knowledge of agile practices from one team to another? No existing research addresses this. We approach the research question from the theoretical framing of knowledge transfer from which we create an initial framework of barriers to knowledge transfer of *agile* practices. The framework is applied to a case study of the knowledge transfer of agile practices between two software teams within the same large organization. In this company Team A had successfully implemented an agile software development process and integrated it with quality management standards including standards from the US Food and Drug Administration (FDA). The focus in the case study is on the knowledge activities to transfer knowledge from Team A and on the adaptation of a similar agile development practices in Team B. Based on the empirical findings from the analysis of the case the framework was extended resulting in a framework for barriers to knowledge transfer of agile development practices.

The remaining paper is organized as follows. In section 2 the literature on knowledge transfer and the barriers is reviewed and the analysis framework on barriers to knowledge transfer is presented. In section 3 the research approach is described. In section 4 the analysis of the knowledge transfer process and the barriers in the case study is presented. Finally the findings are discussed in section 5 and concluded on in section 6.

2 An Initial Framework for Knowledge Transfer of Agile Development Practices

This section presents the theoretical background and the initial framework developed on the basis of the literature.

2.1 Knowledge Transfer and Barriers in General

Within research on knowledge management [21] knowledge transfer in organizations is "the process through which one unit (e.g., group, department, or division) is affected by the experience of another" [6]. It can further be defined as "the application of prior knowledge to new learning situations" [22], [23]. What manifests knowledge transfer are the changes in the knowledge or performance of the recipient unit.

Thus, by measuring the changes in knowledge or performance the effects of the knowledge transfer process can be evaluated [6].

The terms 'knowledge transfer' and 'knowledge sharing' are often used interchangeably in the literature even though they are different [24]. While knowledge sharing mainly refers to the exchange of knowledge between individuals, knowledge transfer also includes higher organizational levels, for example a group or a division [6]. Knowledge transfer is not only about exploiting resources, i.e., knowledge, but also about how to acquire and absorb it to make activities more efficient and effective [24].

The literature distinguishes between two types of knowledge: tacit knowledge and explicit knowledge [25], [26]. Tacit knowledge is deeply rooted in the actions, the experiences as well as the ideas, values and emotions of the individual, which makes it difficult to share with others. Knowledge sharing takes place through people-to-people interactions and dialogue [27], [28]. Explicit knowledge is transmittable in a formal, systematic language [29]. This can be done through codification, i.e., using a people-to-documents approach, where the knowledge is extracted from one person, made independent of that person, and reused for various purposes [27]. Far from all of an organization's tacit knowledge can be made explicit, and far from all explicit knowledge can be documented [11].

Recognizing potential barriers to knowledge transfer is important for the process of understanding which barriers may occur [23] and how to overcome these difficulties of transferring knowledge [22], [30]. Three generic types of knowledge transfer barriers have been identified [23]: individual (e.g., general lack of time, and lack of trust); organisational (e.g., lack of leadership, and restricted knowledge flows); and technical (e.g., lack of integration of systems and processes, and lack of training in new systems). In a Delphi study several barriers were identified at four organisational levels [30]: individual, team, organisation, and inter-organisation, e.g., team climate, organisational relationships.

These theories of knowledge transfer and barriers in general have been used to sensitize our view on knowledge transfer and barriers in the domain of agile software development.

2.2 Knowledge Transfer and Barriers in Agile Software Development

The perspective of knowledge management has been applied to information systems development over the last two decades, e.g., [31], [32], [33], [34], [35], [36], [37], and to explain the potential benefits of agile software development in particular [19] with several proponents of agile processes, e.g., [38] , suggesting that knowledge transfer within teams would be improved, e.g., [15], [16], [17], [39].

The initial framework is based on a literature review of the research on agile software development to identify how agile practices are transferred. Figure 1 depicts the knowledge transfer process schematically to illustrate how knowledge is transferred from one team to another through a deliberate transfer process and how that is slowed, reduced or even hindered by barriers.

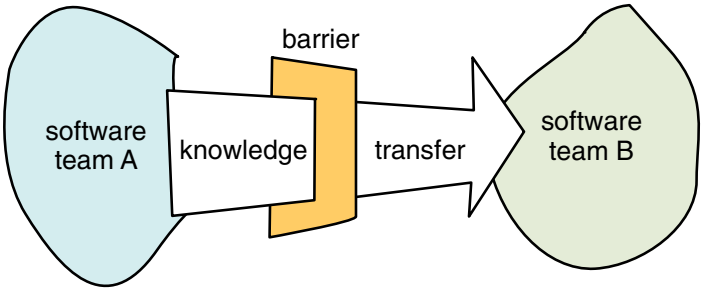

Fig. 1. The knowledge transfer process in agile software development

The framework is further focused on barriers to knowledge transfer in agile software development. From the literature on knowledge transfer and barriers we have thus identified the barriers in table 1. These are all explained in detail in the following.

Table 1. The initial framework of barriers to knowledge transfer of agile development practices

Barrier	Description
Individual skills	Individuals and their skills are central to the outcome of a knowledge transfer process.
Motivation and willingness	The individuals may be more or less motivated to receive knowledge.
Time and resources	Time is a significant factor as the process of identifying and transferring knowledge is very time-consuming.
Organizational culture	Culture is often seen as the key inhibitor of effective knowledge transfer.
Trust	Without trust people are unlikely to share knowledge.
Infrastructure	Integrating an IT system can support the knowledge transfer process.

Individual skills are central to the outcome of a knowledge transfer process, without the right skills, knowledge is not likely to be transferred. If the individual is not part of a social network, it is not possible to access the knowledge [5]. A lack of network ties can result in ignorance on both ends of the transfer. Differences, such as age, gender, educational levels and ethnic background between the participants can hinder the knowledge transfer process, as power issues and problems understanding each other may occur [18], [23]. The skills and emotions of the individual play an important role. The communication skills, both verbal and written, are fundamental to effective and successful knowledge sharing and transfer [23]. Without the ability to communicate the point will not come across and the knowledge will not be transferred. A good coordination between the participants also advances the transfer process [7]. The absorptive capacity of the individual also varies and despite the best intension, these are a boundary for the knowledge transfer [7], [10].

Motivation and willingness can also be a barrier. Individuals may be more or less motivated to receive knowledge from the outside. Lack of motivation may result in

procrastination, passivity, feigned acceptance, sabotage, or maybe rejection of new knowledge [7]. Knowledge sharing activities can neither be supervised nor forced of people; instead, the individuals need to see a value of the knowledge in order to share their knowledge voluntarily [23]. Transferring knowledge also entails people to be able to discard old practices and sustain new ones [7]. Getting people to change their practices is difficult as they easily fall back on well-known routines; it requires a level of willingness from the individual [40].

As the process of identifying and transferring knowledge is very time-consuming, *time* becomes a significant factor. If you lack the time to identify other individuals or groups with specific knowledge, lack the time to share knowledge or lack the time to implement the knowledge in the practice the risk is that the knowledge transfer only succeed partly [23]. Information overload is linked to the time issue; it becomes a barrier because it is difficult to learn anything when there is so much to know in little time [10]. A lack of *resources* to create sharing opportunities is also a potential barrier to a successful knowledge transfer [23].

In the knowledge management literature, *culture* is often seen as the key inhibitor of effective knowledge transfer [41]. It is important that the corporate culture and the organizational design support the knowledge transfer practices [10], [23]. An organizational design in silos creates divisions which only focus on their own accomplishments and tend to safeguard knowledge. Knowledge flows which are restricted in some directions are also a potential organizational knowledge barrier. Large organizational units are unmanageable and make it more difficult to enhance contact and transfer knowledge easily. Lack of formal and informal spaces in which knowledge can be shared are also a potential barrier [23].

Lack of trust can also be a barrier. Without trust people are unlikely to share knowledge. They are reluctant to give away any knowledge if they fear that the knowledge may be misused or taken unjust credit for and they are reluctant to accept knowledge if they lack trust in the credibility of the knowledge or the actors it originates from [23]. Individuals may safeguard their knowledge out of fear that sharing may reduce or jeopardise their job security [42].

Integrating an IT system can support the knowledge transfer process. Such a system however needs an adequate level of informational and technical *infrastructure* (including support and maintenance). It further needs to support the practices of the organization otherwise actors may be reluctant to use such a system [23]. Reluctance may also arise when people are not familiar with the system; communication about the IT-system, training and demonstration of the system is therefore important [23].

3 Interpretive Case Study Approach

This is an interpretive case study [43], [44]. The empirical data were collected over a period of one year (May 2010 to April 2011) in a large pharmaceutical company and specifically with a focus on two software teams (A and B) and the context that they are part of. Data were analysed and validated iteratively through two phases, each focusing on respectively software team A and B. Table 2 summarizes the research approach.

Table 2. Summary oft he research approach

Phase	Focus	Data collection	Duration	Data documentation	Data analysis
1	Software team A	8 qualitative interviews	May 2010	Interviews and observations audio recorded and transcribed	Evaluation of software development process
		6 qualitative interviews	June 2010		
		Document study	June 2010		
		Observations of planning and stand up meetings	August 2010		
2	Software team B	1 qualitative interview	August 2010	Interviews audio recorded and transcribed	Evaluation of the knowledge transfer process
		5 qualitative interviews	October 2010		
		5 qualitative interviews	March 2011		
		Document study	April 2011		

The data collection in Phase 1 was based on an interview guide (A) focusing on how team A developed software and on how the development process meshes agile and traditional development. The interviews were conducted through two iterations and included the process manager, the software architect, software testers and software developers.

The data collection in Phase 2 was based on a separate interview guide (B) focusing on how Team B has adopted the development practices from team A and on evaluating the knowledge transfer process. First, an interview with the software coordinator and the coordinator between software and hardware was conducted. The second and third round included interviews with the two full time software developers, interviews with the process manager from team A (in the role as transfer facilitator) and follow-up interviews with the software coordinator and the coordinator between software and hardware. The interview guide (B) focused on the transfer process, which knowledge mechanisms were used, an evaluation of the outcome and on the barriers hindering the implementation of the transferred knowledge about the software development process.

All interviews were audio recorded, transcribed and coded using Atlas.ti V.6. The analysis of team A's practices mapped the agile software development practices and the knowledge of team A. The interviewees from team A validated the analysis. The analysis of team B mapped the specific barriers hindering the knowledge transfer process, after which the barriers were categorized according to the framework of knowledge transfer of agile practices. Team B and the process manager of team A validated the results.

4 Interpretive Case Analysis

This is a case study of knowledge transfer of agile practices between two software teams within the same pharmaceutical organization. Both software teams were part of projects that developed medical devices with embedded software, which for the final devices entail compliance and subsequent approval by several agencies and their quality standards. The knowledge transfer process was established to pass on the experience of software development from team A to team B.

The software project of software team A was the first project including software development initiated in the pharmaceutical organization. During project A the software practice of the organization was therefore developed and improved gradually. After project A, several other projects with development of embedded software were initiated. All of these projects involved several project groups; pharmaceutical and clinical researchers responsible for the drugs, process engineers responsible for production facilities, mechanical engineers responsible for the product's mechanical parts and hardware engineers responsible for the computer chips controlling the mechanics.

Team A had focused on developing, improving and documenting their software process. They had successfully implemented a software development process which includes agile practices based on Scrum; an agile method that provides guidance for efficient management of projects [38]. Scrum operates with the three roles: Scrum master (responsible for the scrum practice), product owner (the customer) and Scrum team (a self-organizing project team). The development is divided into iterations called sprints. Each sprint is initiated with a sprint planning meeting in which the sprint backlog is agreed on; the sprint backlog is a subset of the product backlog (which the customer is responsible for updating). Every day the scrum team meets for a short status meeting in order to keep track of the progress. After each sprint an increment of the system has been implemented and tested and is demonstrated for the customer.

Teams A and B were different in task, size and complexity. Team A's task was larger and more complex than that of team B. While team A consisted of approximately 30 managers, developers and testers, software team B only consists of three developers, one of these in the role of the software coordinator. Project A had run for several years and was entering the stage of refining the product. So far project B had been focusing on developing prototypes. During the Summer 2010, team B changed its focus from the existing prototype to the overall design, developing two prototypes each focusing on a separate design. In late 2010 the team chose one prototype for further development, and afterwards the focus was on testing this prototype. Despite these differences there was a genuine interest in team B to have the knowledge of agile development transferred. First, the agile development practices were deemed theoretically and practically relevant for team B. Second, the agile processes behind team A were already designed to suit small teams like team B. On this backdrop team B were particularly interested in having the experience with Scrum transferred to them.

4.1 The Knowledge Transfer Activities and Their Effect

Team A had implemented an agile development process. They experimented with the iteration length and settled at 2 weeks, which gave the developers a steady work

rhythm, a dedicated focus and forced the developers to break the tasks down. The iterations were coordinated through daily stand-up meetings, ensuring a shared focus and giving an overview of the progress of the iteration. A scrum board was used to visualize the progress of the iterations; several advantages were associated with the board. Team A had also gained experience with agile estimation and planning and despite an iterative software process, they experienced that estimating tasks is difficult and requires experienced developers. While having implemented several agile practices Team A was, at the same time, able to comply with US Food and Drug Administration's quality regulations and standards. Team A had gained experience in developing software architecture, conducting software tests, peer reviewing and writing documentation. They had furthermore experimented with different tools to support the development process and been through the process of having these tools validated by various quality standards. The data analysis has resulted in summarizing the knowledge and experience of team A in table 3.

The knowledge and experience of team A was deliberately transferred by several means: experience workshops, facilitation, consultancy and adaptation. All of the knowledge transfer initiatives were considered very useful by team B. This was especially the case when the transfer activities followed a person-to-person approach (the experience workshops, the facilitation, the consultancy and the frequent meetings between the process manager of team A and the software coordinator of team B). Table 4 gives an overview of the knowledge transfer activities and their content. Each knowledge transfer activity was identified and their contents determined based on the data analysis. The description of each activity in table 4 is a summary of the descriptions given by the interviewees. The fourth column indicates the number of quotes marked in the coding of the interviews. Most of the knowledge transfer activities took place in the beginning of the project (Spring 2010); even though more activities were planned only a few took place in from September 2010 to March 2011.

Table 3. The knowledge of team A

Knowledge area		Description
Method (Scrum)	Iteration length	Sprints of 2 weeks length give a steady work rhythm.
	Estimation of tasks	Estimation requires experienced developers.
	Product owner	A product owner is important.
	Stand up meetings	Daily stand up meetings give an overview of the progress of the iteration.
	Scrum board	A scrum board is great for visualizing the progress of the iterations.
Software architecture		An overall architecture needs to be defined early in the project.
Software test		Early and iterative tests are recommendable.
Documents		Documents have been designed from scratch.
Peer reviews		Peer reviews are advantageous.
Tools		Using tools from the beginning of the project is recommendable.

Table 4. The knowledge transfer activities and the count of evidence in the coded interviews

Activities	Contents	Description	Quotes	Time
Experience workshop (21)	about Scrum	An agile consultant from software team A gave a two days introduction to Scrum to the developers of software team B. The content was a mix of theoretical descriptions and a description of the Scrum process used by software team A.	11	Spring 2010
	about tools	The process manager of software team A presented their line of tools and the purpose of each tool. One of the employees in charge of the tools of software team A had furthermore held an experience workshop on their tool chain.	8	Spring 2010
	about experiences of project A	The process manager gave a few informal presentations on the experiences of and challenges faced by software team A.	2	Spring 2010
Facilitation (36)	of the Scrum process	For a month an agile consultant from software team A helped, software team B, introduce a Scrum process. The agile consultant was present during these meetings and some of the stand-up meetings. The facilitation process was considered very helpful and enlightening.	24	May and June 2010
	of the software architecture	For 2 weeks the software architect of software team A helped define the software architecture of the system of software team B. He gave short presentations on software architecture and developed a suggestion to the software architecture.	12	May 2010
Consultancy (19)	of the software architecture	The software team held 2 one-day meetings with an external consultant on software architecture. The first meeting took place in June 2010, while the second meeting took place in March 2011. In these meetings the consultant presented some architecture patterns. Software team A has received similar lessons by the consultant.	19	June 2010 and March 2011
Adaptation (38)	of processes and practices	The process manager of software team A and the software coordinator of software team B had since the beginning of the knowledge transfer process scheduled weekly meetings. These meetings were however broken off for a couple of months. The meetings were informal and contained the subjects most urgent to the software coordinator.	15	Initiated start 2010
	of documentation	Software team B had inherited the documents of software team A. These documents not only contained knowledge on how the software was developed in project A, but also represented knowledge on how to construct such documents.	19	Start 2010
	of tools	Employees of software team A had installed a tool chain at software team B.	4	Start 2010

Team B gained much knowledge on Scrum, but had not yet been able to implement these practices in their own team to a level where they had gained their own new experiences. Team B did not take all experience from team A with scepticism and critique. They had run a few test sprints of 1 week length, but did acknowledge the sprint length of 2 weeks as suggested by team A. During these test sprints stand up meetings were held, but were not implemented permanently. Team B had designed a scrum board but the use of it was limited as they had not implemented sprints. Acknowledging the importance of a product owner this role was assigned to the coordinator between the software, hardware and mechanics groups. This role was, however, not filled sufficiently, mainly due to time issues.

Team B experienced difficulties with the estimation of tasks; primarily due to interruptions. They had also acquired own additional knowledge on software test, software architecture and peer reviews, which would become useful for the development of the software for the final device, but as they still were focusing on the software for the prototypes, these processes had not yet come into play. Table 5 summarizes the knowledge of team B.

A comparison of the knowledge of team A and B reveals that the knowledge transfer process only has been partly successful. Team B had not been able to integrate and use all the relevant knowledge in their own software practices. This issue was the subject of several questions during the interviews with team B, and the analysis shows that despite several valid attempts and many good intentions they had not in their own view been sufficiently successful in absorbing the knowledge and experience from team A. The remaining analysis will therefore focus on the barriers of the knowledge transfer process to reveal the underlying causes.

Table 5. The knowledge of Team B

Knowledge		Description
Method (Scrum)	Iteration length	A few test sprints of 1 week length had been run. The sprint length of 2 weeks was however acknowledged.
	Estimation of tasks	Estimating tasks of the test sprints was difficult, mainly due to many interruptions.
	Product owner	The product owner role was assigned to a coordinator.
	Stand up meetings	Daily stand up meetings had not been implemented, but was planning when the sprints were introduced.
	Scrum board	A scrum board was designed, but the use was limited.
Software architecture		The need for an overall architecture was acknowledged, but no time was given to design one.
Software test		The advantages of test-driven development were acknowledged, but the project management had little focus on test at this point.
Documents		Knowledge on how to build documents had been obtained. Many of the documents had been processed.
Peer reviews		Peer reviews were planned for the development of the final device.
Tools		The tools had been installed, but not used yet.

4.2 The Barriers of the Knowledge Transfer of Agile Practices

The analysis showed that several barriers hindered the knowledge transfer of agile practices. The barriers are summarised in table 6. The barriers 'time and resources' and 'organizational culture' were in this case the main barriers and these are therefore described in more detail below. The analysis identified an additional barrier (management style) compared to what was suggested in the initial framework. This barrier will therefore also be described in further detail below.

Table 6. The knowledge transfer barriers and their count of evidence in the empirical data

General barrier	Specific barrier	Description	Quotes
Time and resources (59)	Focus on prototypes	The excessive focus on developing prototypes limits the time to transfer knowledge and adopt agile practices.	34
	Focus on overall design	The transfer of knowledge and the adoption of Scrum were put on hold, due to the focus on developing the overall design.	16
	Software developers not present during the knowledge transfer acts	The software developers were not present during all of the knowledge transfer initiatives. Due to the many urgent support tasks they were not able to allocate the time needed.	9
	Easier in the future	The time issues resulted in an attitude: "it will be easier in the future" among the developers.	5
Organizational culture (64)	The project management do not share organizational culture with software dev.	The project management has a background in mechanics and are having a hard time understanding software development.	33
	Disintegrated organizational cultures	The professional practice of the hardware and mechanics groups is different than the software group.	12
	Focus on mechanics	The deadlines of the entire project depend on the development of the mechanics.	19
Individual skills (15)	Not accustomed to Scrum	None of the software developers had any experience in using Scrum and were not trained in this method beforehand. As the organization is new at software development the software processes are not well-defined.	15
Motivation and willingness (20)	Problems demotivate	The problems experienced during the sprints had a demotivating effect on the software developers, who otherwise were very motivated to implement Scrum.	17
	Management not present during the knowledge transfer acts	The project management was invited to the experience workshops but did not attend. The project management does not prioritize the knowledge transfer and the adoption of Scrum. This is a proof of their lack of motivation and willingness.	3

Table 6. (*Continued*)

Management style (48)	Product owner	The product owner role has been assigned to the coordinator between the software, hardware and mechanics teams. As this is not his official role, he has difficulties allocating the time needed.	31
	Lean	The top management of the organization has introduced lean principles, which has entailed a display of a lean board, on which the deadlines for each project group is displayed. Detailed deadlines are requested, which leads to overlapping information on the lean board and the scrum board.	17

4.3 Time and Resources

'Lack of time' was referred to as a primary reason why the Scrum practices had not been fully adopted, yet. Simultaneously with the knowledge transfer process team B had focused on the overall design and on developing prototypes of the system. These foci highly influenced the daily practice. Team B was expected to support the hardware and mechanics teams with software test script and the majority of these were needed right away, hence difficult to plan in iterations. Supporting the prototypes was almost a full time job for all three software developers.

> "we have two projects in one, I believe that is the greatest challenge, and will be so in the future, we are not able to say, now we have finished that part of the system and now we get to focus on the final product; that is the predominating challenge" (the coordinator between hardware and software in Project B, March 2011).

The software for the final device was to comply with several medical quality standards; but the software for the prototype had not undergone any quality assurance, e.g., writing documentation, peer reviewing of the code and software testing. The development of the final software would therefore become more time consuming and the software team needed to start focusing on developing the software for the final device in order to finish within the final deadline for the overall project. But, team B was left with very little time for the development of the software for the final device and little time to focus on the knowledge transfer and the adoption of new agile practices into their software process.

> "I think it is aggravating that they did not take the time to work towards being ready to start the actual software development" (the process manager from Team A, October 2010).

While focusing on the overall design and on developing the prototypes caused the time pressure, the specific barriers "software developers were not present during the knowledge transfer acts" and "easier in the future" was caused due to the lack of time.

Due to the many urgent support tasks some of the developers were not present during the entire experience workshop and the facilitation of the agile process. An attitude "introducing Scrum will be easier in the future," arose among the software developers as they awaited a decrease in the time pressure.

> "We will not be able to do it before the project has quietened down" (the software coordinator from Team B, October 2010).

The 'lack of time' was therefore both a barrier for the sharing of knowledge and for the implementation of the new knowledge. The fact that some developers did not find the time to attend all of the knowledge transfer activities limited the amount of knowledge shared between Team A and Team B. The fact that the developers did not find time to put the new knowledge into use and were waiting for more time in the future hindered the implementation of knowledge.

4.4 Organizational Culture

As the organization and the project management had very limited experience in software development, they did not share the same organizational culture as the software developers. The project manager had a background in mechanics and had a hard time understanding the challenges of software development. The process of developing mechanical parts is very different from the process of developing software. The disintegrated organizational cultures also appeared as the mechanics and hardware groups lacked the experience in software development. They did, for example, not understand how the support tasks influenced the software team and they did not define the tasks in advance in order for the software team to able to create these as backlog items and include these in the iterations. Getting the hardware and mechanics groups to follow the scrum principles was a great challenge for Team B. An excessive focus on the development of the mechanical parts of the system was furthermore an example of disintegrated organizational cultures in the organization and in the project.

> "As mentioned before, I think there is too much focus on the mechanics; they know how to do that" (the software coordinator from Team B, March 2011).

The differences in the 'organizational culture' became a barrier for both the sharing of knowledge and for the implementation of the knowledge. The management had little interest in attending the knowledge transfer activities and did not prioritize time for the software developers to share or implement knowledge. The priorities of the management had a great influence on the work tasks of the software developers.

4.5 Management Style

The case study revealed a need for a change in management style; implementing the roles of the scrum master and the product owner required a change from the traditional project management. This also became a barrier for the knowledge transfer. The project, including its management was not used to applying these roles and experienced problems using these properly (primarily the product owner role).

As no customer was related to the project, the product owner role had been assigned to the coordinator between the software, hardware and mechanics groups. He had an overview of the whole product and the needs of the three groups. This was however not an official role which let to difficulties allocating the time needed arise.

> "at least we were successful at getting him to attend a few times; it was challenging for him to participate, because he got pulled in from other places as well" (the software coordinator form Team B, October 2011).

The change of 'management style' became a barrier to the implementation of the Scrum practices. The barrier is closely connected to both the barriers of 'time and resources' and 'organizational culture'. The difficulties implementing the product owner role primarily arose do to the lack of time to implement it and time was not allocated as management did not prioritize the product owner and his responsibilities.

4.6 Summary: Knowledge Transfer between Teams A and B

In summary, the analysis shows the transfer of knowledge of agile practices as a difficult undertaking. The environment in which we have studied the case contains a strong imperative to include quality assurance processes (QA) in the software development process as required by the agency, in this case FDA, which has to approve the final product. It is additionally conditioning for the development teams that they are embedded in larger development projects that address the whole product and include teams for hardware development, mechanical development, and clinical development. Neither of these other teams nor the product project work in agile ways and have little knowledge of why that can be both necessary and useful for the software teams.

Table 7. Knowledge transfer between Team A and Team B

Team A Knowledge areas	Knowledge transfer activities	Barriers	Team B Knowledge areas
Method (Scrum): iteration length, estimation of tasks, product owner, stand-up meetings, Scrum board Software architecture Test Peer review Documents Tools	Experience workshop (21) Facilitation (36) Consultancy (19) Adaptation (38)	Skills (15) Motivation (20) Time and resources (59) Organisational culture (38) Management style (48) Trust (0) Infrastructure (0)	Method (Scrum): iteration length, estimation of tasks, product owner, stand-up meetings, Scrum board Software architecture Test Peer review Documents Tools

This case study of knowledge transfer of agile software practices showed that, even though knowledge was transferred by several means, the knowledge transfer process was only partially successful. The knowledge of Team B covers the same knowledge areas, but is not the same as the knowledge of Team A and is not at the same level of understanding own experience. Team B had retrieved knowledge of the experiences and software practice of Team A, but had not been able to implement the knowledge in their software practice, yet. Table 7 provides a generalised overview of knowledge areas, transfer activities and barriers to knowledge transfer.

5 Discussion

The knowledge transfer literature has acknowledged the importance of focusing on barriers [7], [23]. This claim is supported by this case study, in which several barriers influence the effects of knowledge transfer to a high degree. The initial framework of barriers for knowledge transfer of agile software practices (table 1) has been applied to the case study as suggested in figure 1. This provided us with a fuller description and understanding of the potential barriers. The case study showed how each barrier specifically emerged and affected the results.

The case has also shown that in order to better understand the potential barriers it is necessary to include specific barriers relevant to the knowledge transfer and barriers of agile software development. No other research has done that. In this discussion we will therefore address the barriers found in the case and relate these both the literature on knowledge transfer and the literature on adoption of agile development methods.

5.1 Potential Barriers to Knowledge Transfer of Agile Development

The case study and the literature show several potential barriers; but the support in the literature varies, and it seems specific to the knowledge transfer of agile development that ‘management style’ is a barrier. The barriers have support in the literature and in the case as follows.

In the literature on knowledge transfer the barriers ‘individual skills’, ‘motivation and willingness’ and ‘time and resources’ are described as potential barriers across all transfer activities [5], [7], [23]. In the literature on agile development there is support for the claim that these barriers can hinder the implementation agile development [45], [46], [47]. When the agile development literature is viewed from the perspective of knowledge transfer there is concurrence. The case study showed how all three of these barriers have affected the transfer of knowledge from team A to team B. Hence, there is even stronger support for the claim.

For ‘time and resources’ in particular it is relevant to consider the knowledge contents. The literature acknowledges that both knowledge transfer [23] and the adoption of agile practices is time consuming [38]. It is difficult to learn anything when there is so much to know in so little time [10]. The knowledge transfer of agile development in team B therefore becomes very time consuming. The adoption process of team A has taken several years and so far the adoption process of team B

has been on-going for almost a year without much progress. Getting the project management (and the other teams) to allocate the time needed for the knowledge transfer and adoption process is important in order to ensure a successful transfer and adoption of the agile practices. A lack of time and resources can therefore be a substantial barrier. If there is no time and no resources to identify knowledge, the transfer will not be initiated. In order to be able to share knowledge between two parties 'time and resources' are needed [23]. In the case study 'time and resources' was to some degree a barrier to the sharing of knowledge as not all of the developers and the management of team B had time to attend the knowledge sharing activities. 'Time and resources' mainly became a barrier for implementation of the acquired knowledge; that time to do this is essential is supported by [38].

In the literature on knowledge transfer the barrier 'organizational culture' can hinder the identification of knowledge[41] and the sharing of knowledge [23]. In the literature on agile development this barrier is also claimed to hinder introducing agile development in an entire organisation [48]. This case study supports this claim as management and the other project groups did not share the same organizational culture as the software developers. The adoption of the agile practices such as the sprints were hindered or at least very difficult. It is further claimed in the literature on agile development that the organisational culture needs to change from policy and procedure-based to that of freedom of development and management by team members [49]. The case study supported this as it showed how the organizational culture was to a large degree a barrier to both sharing of knowledge of agile development and to implementing the experience from team A in team B.

The literature on knowledge transfer does not mention 'management style' as a barrier. The literature on agile development suggests that the transition from traditional management to an agile self-organizing team proposes many challenges [3], [47]. The case study supports this by showing how team B had difficulties implementing the product owner role.

The literature on knowledge management pointed that 'trust' and 'infrastructure' as potential barriers for the transfer of knowledge; but neither the literature on agile development nor the case study supported this. They are however still potential barriers and need to be included in the final framework of potential barriers to knowledge transfer of agile development.

The literature on agile development advocates agile methods to alleviate problems with knowledge sharing. Much of this literature is concerned with knowledge sharing within a development team, e.g., [15], [16], [17], [38], and example of a practice supporting knowledge sharing is pair programming [50]. Knowledge sharing is never easy even within an agile software team [19], [20], [51]. The present case study addresses not the knowledge sharing within a team but between teams. The research literature on knowledge transfer between teams is sparse; and the exceptions are the research on use of post-mortem evaluations and a recent case study. Post-mortem evaluation in agile development is an attempt to systematise learning from one agile project to another [52] through explicating experience and knowledge. Such transfer mechanisms would go hand in hand with what we have seen in our case study – it is very close to a post mortem that was performed by team A in the process of

transferring to team B. The main difference was that the evaluation was directed at team B and in a meeting. A minor difference is that the original intention with their ideas was to improve internally and between iterations within the same team as well – this was not an interest for teams A and B. The techniques in post mortem evaluation are likely to be useful also for the transfer of knowledge between teams and perhaps a systematic technique can overcome some of the barriers. This last claim is not supported by our case study.

The case study by [18] is interesting because it is based on a framework of knowledge transfer and it specifically addresses how this led to several benefits. The case study is brief, but it does convey experiences with forming deliberate knowledge transfer through moving experiences developers between teams. While teams A and B did not have that opportunity it seems likely that the tacit nature of knowledge of agile development can benefit from this approach to knowledge transfer and that this is fully aligned with the knowledge sharing strategies within agile teams [19].

5.2 The Resulting Framework for Knowledge Transfer of Agile Practices

The framework presented in table 1 has been adapted and the resulting framework is presented in table 8. This framework presents the potential barriers for knowledge transfer of agile practices.

Table 8. The resulting framework of barriers to knowledge transfer of agile development

	Knowledge transfer literature	Agile development literature	Case study
Individual skills	√	√	√
Motivation and willingness	√	√	√
Time and resources	√	√	√
Organisational culture	√	√	√
Management style		√	√
Trust	√		
Infrastructure	√		

The resulting framework (table 8) relates to the maturation of knowledge in organisations in general. A framework of knowledge maturation [53] suggests to align knowledge management and organisational learning and it is argued that reflection is one of the major mechanisms that leads to maturing of organisational know-how. They further found that reflection improves the maturity of organisational processes and that it contributes to the development of shared know-how, organisational best

practice, and standardisation of work processes. Reflection has previously been applied to explain learning in software development [54], [55], [56]. This line of research tells us that the above framework should be seen in this light as well. The implication is that while there may be barriers to the transfer of knowledge on agile development we should not expect that the removal of these barriers will lead to a high level of adoption of agile development as this will also require reflection of own practices and perhaps even reflection-in-action [55], [57] as part of agile development.

The issues of knowledge transfer of agile development is related to the issues with applying development methods (in general and not only agile methods) in practice[58], [59], [60], [61], [62], [63], [64], [65], [66]. It is commonly claimed in the research literature that software development methods are never simply applied in practice, but they may inform practitioners nevertheless, and a reasonable way to understand this new form is as a method-in-action [61], [62] as this also applies to agile methods [67]. These differences between methods and practice persist in all of software development and the framework of barriers (table 8) should be viewed on this backdrop.

The issues of knowledge transfer barriers further relates to the adoption of software development methods and similar process innovations [58], [68], [69], [70]. Team B is trying to learn from the experience of team A and in the process they also try shape their own method-in-action. It emerges gradually to them that they have to adapt team A's experience to their own situation and that very much resembles the 'emergent method' based on innovation theory [58] that comes about over time through the interaction between structural influences, actions of individuals, as well as the knowledge of agile development they are seeking to assimilate.

To team B any agile development practice is a process innovation that they wish to evaluate and possibly employ in their own project. As such it relates to how development practices are formed and can be explained in a NIPO grid [68]. The NIPO grid sees practice as a multidimensional concept and consists of two dimensions: the intended scope (how widespread the practices are intended to be) and the actual scope (how widespread the actual use of a practice is). The scopes can vary from not enacted at all (N), enacted by individuals (I), being enacted at project level (P) or enacted throughout the organization (O). Looking only at team B they express a desired change with intended use of several agile practices at P (team level) and almost all actual uses at N (no use) and additionally that the intended use never included O (the whole organization, but only team B). The grid framework then specifies four possible change paths: (1) tacitly through mutual learning and interaction, (2) as other people in a project start using it (3) because it is innovated in a project, or (4) as an alteration to a formalized method [68]. Trying to learn from team A makes team B relying primarily on the third path. Looking back at team A they employed all four paths in changing their actual use to P. Team B had gradually begun applying paths 1 and 2 as well. Relative to the NIPO grid the resulting knowledge transfer framework details path 3 and how to design transfer activities and in particular it add much insight to which barriers might occur on path 3.

6 Conclusions

This case study of knowledge transfer of agile software development shows that it is not straightforward. Many potential barriers can affect the knowledge transfer. In this paper we have developed a framework of barriers to knowledge transfer of agile software development. The framework has proven its relevance in explaining the case, as it provides a better understanding of the barriers to knowledge transfer of agile software development and hereby gives a greater possibility to choose better counter-measures to overcome these barriers.

We thus suggest that for both practitioners and researchers our framework will be relevant and potentially useful. Having identified the barriers it is important to choose the right counter-measure overcome the barriers.

The research has been conducted as a case study and we have emphasised how we have collected the data and how we have analysed these. However, there are general limitations of the case study approach and in particular limits to the generalisations that can be made[71] we do find that much can also be learned from a single case study[72].

As for future research on this topic we suggest that two issues in particular will be interesting: (1) To address the development of a repertoire of strategies for knowledge transfer between agile teams and in particular counter-measures acting against barriers as the existing literature only contains little advice on how to deal with the barriers. (2) To address the need for more generalizable research perhaps working in more detail with the most important barriers, and to this end we find that more case studies will be effective, but also action research where researchers will work in close collaboration with a client software organisation.

References

1. Conboy, K.: Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. Information Systems Research 20, 329–354 (2009)
2. Dingsøyr, T., Dybå, T., Moe, N.B.: Agile software development: Current research and future directions. Springer, Heidelberg (2010)
3. Dybå, T., Dingsøyr, T.: Empirical Studies of Agile Software Development: A Systematic Review. Information and Software Technology 50, 833–859 (2008)
4. Senapathi, M.: Adoption of software engineering process innovations: The case of agile software development methodologies. In: Sillitti, A., Martin, A., Wang, X., Whitworth, E. (eds.) XP 2010. LNBIP, vol. 48, pp. 226–231. Springer, Heidelberg (2010)
5. O'Dell, C., Grayson Jr., C.J.: If only we knew what we know: The transfer of internal knowledge and best practice. The Free Press, New York (1998)
6. Argote, L., Ingram, P.: Knowledge Transfer: A Basis for Competitive Advantage in Firms. Organ. Behav. Hum. Decis. Process. 82, 150–169 (2000)
7. Szulanski, G.: The Process of Knowledge Transfer: A Diachronic Analysis of Stickiness. Organ. Behav. Hum. Decis. Process. 82, 9–27 (2000)
8. Carlile, P.R.: A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development. Organization Science 13, 442 (2002)
9. Carlile, P.R.: Transferring, Translating, and Transforming: An Integrative Framework for Managing Knowledge Across Boundaries. Organization Science 15, 555 (2004)

10. Lyytinen, K., Robey, D.: Learning Failure in Information Systems Development. Information Systems Journal 9, 85–101 (1999)
11. Lindvall, M., Rus, I.: Knowledge Management in Software Engineering. IEEE Software 19, 26–38 (2002)
12. Faraj, S., Sproull, L.: Coordinating Expertise in Software Development Teams. Management Science 46, 1554–1568 (2000)
13. Desouza, K.C.: Barriers to Effective use of Knowledge Management Systems in Software Engineering. Communication of the ACM 46, 99–101 (2003a)
14. Desouza, K.C.: Facilitating Tacit Knowledge Exchange. Communications of the ACM 46, 85–88 (2003b)
15. Chau, T., Maurer, F., Melnik, G.: Knowledge sharing: Agile methods vs. tayloristic methods. In: Proceedings of the Twelfth IEEE International Workshops on Anonymous Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2003, pp. 302–307. IEEE (2003)
16. Maurer, F., Melnik, G.: Agile methods: moving towards the mainstream of the software industry. In: Anonymous Proceedings of the 28th International Conference on Software Engineering, pp. 1057–1058. ACM (2006)
17. Melnik, G., Maurer, F.: Direct verbal communication as a catalyst of agile knowledge sharing. In: Anonymous Agile Development Conference, pp. 21–31. IEEE (2004)
18. Tellez-Morales, G.: XP practices: A successful tool for increasing and transferring practical knowledge in short-life software development projects. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) XP 2009. LNBIP, vol. 31, pp. 155–160. Springer, Heidelberg (2009)
19. Joshi, K., Sarker, S., Sarker, S.: Knowledge Transfer within Information Systems Development Teams: Examining the Role of Knowledge Source Attributes. Decis. Support Syst. 43, 322–335 (2007)
20. Sarker, S., Nicholson, D., Joshi, K.: Knowledge Transfer in Virtual Systems Development Teams: An Exploratory Study of Four Key Enablers. IEEE Transactions on Professional Communication 48, 201–218 (2005)
21. Alavi, M., Leidner, D.E.: Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundation and Research Issues. MIS Quarterly 25, 107–136 (2001)
22. Riege, A.: Actions to Overcome Knowledge Transfer Barriers in MNCs. Journal of Knowledge Management 11, 48–67 (2007)
23. Riege, A.: Three-Dozen Knowledge-Sharing Barriers Managers must Consider. Journal of Knowledge Management 9, 18–35 (2005)
24. Liyanage, C., Elhag, T., Ballal, T., et al.: Knowledge Communication and Translation–a Knowledge Transfer Model. Journal of Knowledge Management 13, 118–131 (2009)
25. Polanyi, M.: The tacit dimension. Anchor Day (1966)
26. Polanyi, M.: Personal knowledge. University of Chicago Press (1962)
27. Hansen, M.T., Nohria, N., Tierney, T.: What's Your Stategy for Managing Knowledge. Harv. Bus. Rev., 106–116 (March-April 1999)
28. Hansen, M.T.: Knowledge Networks: Explaining Effective Knowledge Sharing in Multiunit Companies. Organization Science 13, 232–248 (2002)
29. Nonaka, I.: A Dynamic Theory of Organizational Knowledge Creation. Organization Science 5, 14 (1994)
30. Sun, P.Y., Scott, J.L.: An Investigation of Barriers to Knowledge Transfer. Journal of Knowledge Management 9, 75–90 (2005)
31. Althoff, K., Bomarius, F., Tautz, C.: Knowledge Management for Building Learning Software Organizations. Inf. Syst. Front. 2, 349–367 (2000)
32. Nielsen, P.A., Tjørnehøj, G.: Social Networks Analysis in Software Process Improvement. Software Process Improvement and Practice 22, 33–51 (2010)

33. Mathiassen, L., Pedersen, K.: Knowledge Creation and Sharing in a Systems Development Project. International Journal of Business Information Systems 3, 1–20 (2008)
34. Nørbjerg, J., Elisberg, T., Pries-Heje, J.: Experiences from Using Knowledge Networks for Sustaining Software Process Improvement. In: Anonymous Experiences from Using Knowledge Networks for Sustaining Software Process Improvement, Rio de Janerio, Brazil (2006)
35. Nielsen, P.A., Kautz, K.: Software processes & knowledge: Beyond conventional software process improvement. Software Innovation Publisher, Aalborg (2008)
36. Crawford, B., Castro, C., Monfroy, E.: Knowledge management in different software development approaches. In: Yakhno, T., Neuhold, E.J. (eds.) ADVIS 2006. LNCS, vol. 4243, pp. 304–313. Springer, Heidelberg (2006)
37. Kjærgaard, A., Nielsen, P.A., Kautz, K.: Making Sense of Software Project Management. Scandinavian Journal of Information Systems 22, 3–26 (2010)
38. Schwaber, K., Beedle, M.: Agile software development with scrum, vol. 18. Prentice Hall (2001)
39. Ryan, S., O'Connor, R.V.: Development of a Team Measure for Tacit Knowledge in Software Development Teams. J. Syst. Software 82, 229–240 (2009)
40. Gupta, K.S.: A Comparative Analysis of Knowledge Sharing Climate. Knowledge and Process Management 15, 186–195 (2008)
41. McDermott, R., O'Dell, C.: Overcoming Cultural Barriers to Sharing Knowledge. Journal of Knowledge Management 5, 76–85 (2001)
42. Michailova, S., Husted, K.: Knowledge-Sharing Hostility in Russian Firms. Calif. Manage. Rev. 45, 59–77 (2003)
43. Walsham, G.: Interpretive Case Studies in IS Research: Nature and Method. European Journal of Information Systems 4, 74–81 (1995)
44. Walsham, G.: Doing Interpretive Research. European Journal of Information Systems 15, 320–330 (2006)
45. Esfahani, C., Cabot, J., Yu, E.: Adopting agile methods: Can goal-oriented social modeling help? In: Anonymous Fourth International Conference on Research Challenges in Information Science (RCIS), pp. 223–234. IEEE, Nice (2010)
46. Cohn, M., Ford, D.: Introducing an Agile Process to an Organization [Software Development]. Computer 36, 74–78 (2003)
47. Boehm, B., Turner, R.: Management Challenges to Implementing Agile Processes in Traditional Development Organizations. IEEE Software 22, 30–39 (2005)
48. Svensson, H., Host, M.: Introducing an agile process in a software maintenance and evolution organization. In: Anonymous Ninth European Conference on Software Maintenance and Reengineering, CSMR 2005, pp. 256–264. IEEE, Manchester (2005)
49. Misra, S.C., Kumar, V., Kumar, U.: Identifying some Critical Changes Required in Adopting Agile Practices in Traditional Software Development Projects. International Journal of Quality & Reliability Management 27, 451–474 (2010)
50. Beck, K., Andres, C.: Extreme programming explained: Embrace change. Addison-Wesley Professional (2004)
51. Sarker, S., Sarker, S.: Exploring Agility in Distributed Information Systems Development Teams: An Interpretive Study in an Offshoring Context. Information Systems Research 20, 440–461 (2009)
52. Dingsøyr, T., Hanssen, G.K.: Extending agile methods: Postmortem reviews as extended feedback. In: Henninger, S., Maurer, F. (eds.) LSO 2003. LNCS, vol. 2640, pp. 4–12. Springer, Heidelberg (2003)

53. Kump, B., Knipfer, K., Pammer, V., et al.: The role of reflection in maturing organizational know-how. In: Reinhardt, W., Ullmann, T.D., Scott, P., Pammer, V., Conlan, O., Berlanga, A. (eds.) Proceedings of the 1st European Workshop on Awareness and Reflection in Learning Network, pp. 30–45 (2011)
54. Mathiassen, L., Purao, S.: Educating Reflective Systems Developers. Information Systems Journal 12, 81–102 (2002)
55. Mathiassen, L.: Reflective Systems Development. Scandinavian Journal of Information Systems 10, 67–117 (1998)
56. Heiskanen, A., Newman, M.: Bridging the gap between information systems research and practice: the reflective practitioner as a researcher. In: Anonymous Bridging the Gap between Information Systems Research and Practice: The Reflective Practitioner as a Researcher, ICIS 1997: Proceedings of the Eighteenth International Conference on Information Systems, pp. 121–132. Association for Information Systems, Atlanta (1997)
57. Schön, D.A.: The reflective practitioner: How professionals think in action. Basic Books, New York (1983)
58. Madsen, S., Kautz, K., Vidgen, R.: A Framework for Understanding how a Unique and Local IS Development Method Emerges in Practice. European Journal of Information Systems 15, 225–238 (2006)
59. Avison, D., Fitzgerald, G.: Information systems development: Methodologies, techniques and tools. McGraw Hill, Maidenhead (2003)
60. Wynekoop, J., Russo, N.: Studying System Development Methodologies: An Examination of Research Methods. Information Systems Journal 7, 47–65 (2003)
61. Fitzgerald, B., Russo, N.L., O'Kane, T.: Software Development Method Tailoring at Motorola. Commun. ACM 46, 64–70 (2003)
62. Fitzgerald, B., Russo, N.L., Stolterman, E.: Information systems development: Methods in action. McGraw-Hill (2002)
63. Nandhakumar, J., Avison, D.E.: The Fiction of Methodological Development: A Field Study of Information Systems Development. Information Technology & People 12, 176–191 (1999)
64. Russo, N.L., Stolterman, E.: Exploring the Assumptions Underlying Information Systems Methodologies. Information Technology & People 13, 313–327 (2000)
65. Fitzgerald, B.: An Empirical Investigation into the Adoption of Systems Development Methodologies. Information & Management 34, 317–328 (1998)
66. Nielsen, P.A.: Learning and using methodologies in information systems analysis and design. Lancaster University, Lancaster (1990)
67. Fitzgerald, B., Hartnett, G., Conboy, K.: Customising Agile Methods to Software Practices at Intel Shannon. European Journal of Information Systems 15, 197–210 (2006)
68. Larsen, E.Å., Päivärinta, T., Smolander, K.: A Model for Analyzing Changes in Systems Development Practices. Journal of Information Technology Theory and Application (JITTA) 13, 3 (2013)
69. Mustonen-Ollila, E., Lyytinen, K.: Why Organizations Adopt Information System Process Innovations: A Longitudinal Study using Diffusion of Innovation Theory. Information Systems Journal 13, 275–297 (2003)
70. Mustonen-Ollila, E., Lyytinen, K.: How Organizations Adopt Information System Process Innovations: A Longitudinal Analysis. European Journal of Information Systems 13, 35–51 (2004)
71. Lee, A.S., Baskerville, R.L.: Generalizing Generalizability in Information Systems Research. Information Systems Research 14, 221–243 (2003)
72. Flyvbjerg, B.: Five Misunderstandings about Case-Study Research. Qualitative Inquiry 12, 219--245 (2006)

Digital Living at Home - User Voices about Home Automation and a Home-Keeping Design Discourse

Ellen Christiansen and Pernille V.K. Andersen

Aalborg University, Dept. of Communication, Nyhavnsgade 14, DK-9000 Aalborg Denmark
{ech,pvka}@hum.aau.dk

Abstract. Does living with digital technology inevitably lead to digital living? Users talking about a digital home control system, they have had in their homes for eight years, indicate that there is more to living with digital technology than a functional-operational grip on regulation. Our analysis of these user voices has directed us towards a 'home-keeping' design discourse, which opens new horizons for design of digital home control systems by allowing users to perform as self-determined controllers and groomers of their habitat. The paper concludes by outlining the implications of a 'home-keeping' design discourse.

Keywords: interface-design**,** digital living, home control, automation, design discourse, housekeeping, home-keeping, user-centered design, collective resource approach.

"Design strategies that go against the ecological wisdom of a culture are likely to fail" [1: 21]

1 Introduction

Energy use in buildings accounts for almost 40% of all CO_2 emissions in the EU and the building sector is identified as providing the largest potential for CO_2 reduction. In order to unfold both the potential of buildings' energy savings and to handle fluctuating energy, user needs and behavior have to be taken into consideration. The question is how. We have analyzed homeowners' talk, and this analysis leads us to introduce an alternative discourse for communication about energy consumption in private households, which instead of a product focus on comfort, has a process focus on housekeeping – or as we have come to call it 'home-keeping - a perspective in contrast to the 'techno-optimist perspective' presented in formulations by product vendors of home automation[1].

Digital living – what does it mean? Terms like 'digital living' invokes certain discourses and silences others. Each discourse allows some problems to be visible, while they leave other problems in the dark. So, when demonstration projects on

[1] See eg. this website http://www.schneider-electric.com.my/sites/malaysia/en/solutions/energy_efficiency/quick-navigation/home-control-system.page

M. Aanestad and T. Bratteteig (Eds.): SCIS 2013, LNBIP 156, pp. 40–52, 2013.

low-energy buildings show that users do not understand the design intentions and functionality of the new building systems [2] [3] [4] and have serious difficulty in controlling the technologies, it might be because the discourse of the digital products does not comply [sufficiently] with users' everyday living, thinking and talking. Consequently, despite all technical effectiveness, users may live with a poor indoor environment and increased energy use, which result in the buildings neither meeting their expected performance, as planned by the constructors, nor the conservation goals that was hoped for at the political level in the overall sustainability agenda. We see this contradiction as what Alexander [5], describes as a clash between native, situated, unconscious design, where design choices are guided by *"patterns of myth, tradition, and taboo which resists willful change"*[5: 48], and science-based self conscious design, where choices of change are guided by ideas and materials arriving as standard solutions, coming from far away into the actual setting of use, thereby creating a communication problem between the problem setting of design and the solution setting of use – a contradiction experienced every day, and well described in CSCW contexts [6] [7] [8] although mostly from a developer perspective in a 'design-*for*-users'-discourse.

Here our aim is to establish a 'design-*with*-users'-discourse meant for the design-and-negotiation table, where stakeholders in building processes make decisions about interfaces to home control systems. In our case the stakeholders will typically be utility-representatives, vendor-representatives, builder-representatives, and project engineers. Our goal is to bring life to their imagination about users and use, in a way so that the agreements take actual user practice and actual user experience into consideration, in a way which help the stakeholders to explain, later, when they talk to their respective support bases back home, how users, through their everyday practice, make sense of experiences of use. We do this in our capacity of being part of a Danish research initiative to provide an inclusive understanding of user practices among utilities, product developers and building contractors. Our focus is improving communication among stakeholders about how sense-making happens, when being confronted with new energy efficient technologies[2]. The reflections we present here are meant to open our research into communication tools for designers, builders, maintenance people, and politicians, when dealing with private households' attitudes to when, where and how much they want to engage in utility-conservation behavior.

Within design anthropology and related research various ways of conceptualizing local sense-making have developed. Here we apply Bakhtin's concept of appropriation of language and tools as it has been developed by J. Wertsch [9]. Bakhtin, a Russian semiotician, developed the concept of appropriation to capture the process of utterances and tools acquiring meaning:

"Language is not a neutral medium that passes freely and easily into the private property of the speaker's intentions; it is populated –overpopulated– with the intentions of others. Expropriating I, forcing it to submit to one's own intentions and accents, is a difficult and complicated process... As a living, socio-ideological

[2] UserTEC - User Practices, Technologies and Residential Energy Consumption funded by the Danish council for strategic research by 17,2 mio. kr. for 2013-2018.

concrete thing, as heteroglot opinion, language, for the individual consciousness, lies on the borderline between oneself and the other... The word in language is half someone else's. It becomes one's "own" only when the speaker populates it with his own intentions, his own accent, when he appropriates the word, adapting it to his own semantic and expressive intention. Prior to this moment of appropriation, the word does not exist in a neutral and impersonal language... but rather it exists in other people's mouths, in other people's contexts, serving other people's intentions; it is from there that one must take the word, and make it one's own" [10: 294]

Bakhtin, influential in the field of Human Computer Interaction (Spinuzzi [11], Bødker and Christiansen [12]), maintained that an expression in a living context of exchange, like the data we present below, is the main unit of meaning (not abstract sentences out of context), and is formed through a speaker's relation to Otherness (other people, others' words and expressions, and the lived cultural world in time and place). Subjective by nature and definition, meaning, to the extent that it can be shared, is shared in combinations of doing and talking [10].

An utterance in context is always already embedded in a history of expressions, the hermeneutic/semiotic analysis of which can reveal a chain of ongoing cultural and political moments, a dialogic process between user, artifact and situation. Bakhtin coined the term 'appropriation' to designate an action by the user (using the artifact or word), overcoming a socio-cultural and physical resistance. This 'over-coming' is a gradual making sense going from (1) anticipation, (2) initial familiarity, (3) development of repertoires of routines and the (4) development of new forms of use [9: 54].

In Scandinavia we have a prominent tradition in computing for taking linguistic expressions in context seriously [13] [14] [15] [16] [17] [18] [19] [20] [21]. This is especially important in design, where the power of naming things is real power [22].

In Section 2 we will investigate utterances from people who live with an automatic home control. We do it through an 'appropriation-lens'.

In Section 3 we deal with a number of insights about how users make sense of living with a digital home control. In section 4 we anchor our reflections in the concept of home-keeping, in a design discourse developed out of the collective resource approach of Scandinavia. Bjerkness and Bratteteig [20] gives an overview of the research which has proved the gain of involving users in design, in cases where the development of shared meaning is needed: in combinations of doing and talking. In the collective resource approach the shared meaning is found through the joint effort of producing prototypes while talking and conceptualizing.

In the concluding section 5 we sum up our argumentation for a home-keeping design discourse as a way of bringing user-living into the design process from the beginning.

2 Voices of Inhabitants Talking about Everyday Life with Home Control

In this section we illustrate our overall point of this paper that 'living' - digital or not – is a matter of peoples' sense-making, as much as it is about engineering efficient

solutions. We do this through a follow up study on a study conducted by Kanstrup and Christiansen [23] in 2005 investigating eight households' innovative potential regarding design of home control displays. This study was initiated by a pre-study, where three couples living in homes with IHC® (Intelligent Home Control) were visited. All couples had newly moved into high profile apartments. The building company had offered them a discount on installation of IHC® as part of a research project conducted in collaboration between a vendor of electricity controls, the building company, and the national building development board in Denmark. For the residents, the gain was cheap installation of an IHC® software package, a so-called 'comfort' and 'control' system. The software was offered at three levels of complexity, and the informants had chosen the basic package because of its low prize. None of the couples had fully unpacked their moving boxes at the time, and two of the three had not yet put up more than a few lamps, they had not determined where to place the TV, and were yet to locate their home office environment. They were asked about IHC® as a technology, the history of how they ended up deciding to buy this technology, and about their expectations as to what the technology might do for them once they were settled in their new environment. Kanstrup & Christiansen summarized the interesting differences between the three couples according to the activity theory framework in terms of operations, actions and overall ideas:

- Household 1: Operations characterized by pointing at the technology. Actions characterized by description of what they can do and excuses like "we have not got around to the technical yet". Motive verbalized with term like in the "sales speech" talking about the advantages of the technology identical to the arguments found in the manual for the IHC® system.
- Household 2: Operations characterized by pointing. Actions characterized by descriptions of the limits of the system e.g. "in principle yes, but you cannot manage very much from here, in reality we can not connect TV, Internet, and phone". Motivation characterized by descriptions of what the modern family need, and how they feel like "not the target group": "we don't need to call our oven from work". This couple ridiculed some of the system's facilities e.g. that you can program the plugs, so that you from the bedside can light up all the way to the bathroom at night, which they called "the pee-route", a facility they found superfluous.
- Household 3: Operations characterized by showing remote controls and pulling connectors. Actions characterized by descriptions of how they have set up the system using their intuition. Motive characterized by descriptions of their ideas of the future home, to which the IHC® system is hoped to be an answer to e.g. "I would like to be able to sit with my computer on the terrace", and "it has to be energy-efficient".

Now, eight years later, in the spring of 2013, as part of the UserTEC project, these authors re-visited household 1 and 3. When asked, household 2 declined our visit, kindly stating in their e-mail reply that *"we do not believe we can contribute, since we have in no way gotten involved with this IHC-project, never went to the product*

homepage, or explored the control-possibilities offered by the IHC® system. Why? We can only repeat what we said at your first visit: we never found that this was an interesting project."

Below we present key expressions from our interviews with the two remaining households. One took place at the dinner table, and the other, while one of the household members walked us around the house. Household 1, said at the re-interview, that they had done nothing to set up the control system themselves, and after these eight years they were still confused about which switch to hit when. Somewhat surprisingly household 3, who had high hopes eight years ago, had done nothing of what they imagined. They could not remember their optimism from back then, had never tried to program the system, but were agnostic about apparent malfunction.

These three households seem to have not appropriated their IHC® systems at all, although they still live with them. None of the households had familiarized with the technology, let alone developed routines or new forms of use. In that sense the control device they talked about was not appropriated. All it meant to the informants was annoyance – or? In fact, we speculate, it also meant an attack on their general feeling of being in control of their home. It did not only make them indifferent, it actually made them feel having their power of control diminished. In what follows we illustrate this as expressed by the user voices:

"Even though I have lived here for many years by now, I still find myself running around and pushing the wrong buttons, because one haven't for real… And what bothers me is that one never know whether it is on or off this one (pointing at the switch) And often by accident I push this one instead of this one (pointing at two different icons on the same switch) And that means that I can not turn on the television, then, I will have to go to … , and then I figure that .., that this means that it (the switch) is shot off. And so on and so forth, it's very annoying." (Household 1)

"Well, it is amazing, that we after eight years don't know how to turn on the light in the hallway" (Household 3)

"What I think that I can do is something on this (pointing at a switch at the entrance door) Here I can shut it all off (all the lighting) (…) But I think that it's this one, no I'm not sure, it's this one (points at different icons on the IHC® switch). Oh, it was not this one (pushes one of the icons), then I will try this one (pushes at a new icon). It wasn't this one either (pushes a new icon), then it has to be this one (pushes another icon)(…) I don't know if this will shut off the whole thing, but this light at least (the light in the hallway and the living room). But there is some place, where I can shut it all off. Aaghh, it was this one (pushes the last icon on the switch) And that makes sense. That (icon) was a house with some kind of man" (Household 3)

Fig. 1. HCI® switch and the owner's home decoration

The fact that they do not appropriate the IHC® control technology does however not mean that they are not interested in having an understanding of what they call the 'logic' of operation, which we, following the definitions of Activity Theory [24] interpret as a state where operation has become a tacit routine not requiring focal attention.

"I was here at the time where they programmed our light. Because of that there is a certain, because I corrected the guy, because he hadn't thought of, I don't know what he had thought of. But I have in any case made a systematic, where okay, the first switch is controlling this light (in the living room), and the middle one is in control of the light in the kitchen, and the last one in the hallway. In that sense there is a logic. And it is important as well, that the people who install the units, have thought through the logics" (Household 3)

What, however, also became clear was that going about switching on and of light was part of a larger and very meaningful process, namely that of grooming and caring for their habitation:

"We have tree power outlets, and the last one of them is connected in a way so that it can be controlled centrally. And I was considering ... But I find that it is very nice to stroll around and then push that one, 'now there will be light' by pushing this (Points at different lamps in the living room) It's also because whether or not I turn on all lamps is dependent on how much light or how much darkness there is." (Household 3)

In this analysis, what we take away is that the anticipation, although negative, give a hint about what these users want regarding digital home control:

- they do want to be in control of their environment, but if they do not see a way to take immediate action, they live with handing over the action to the system and having their power of control diminished.
- they want there to be *their* logic behind a function.
- they do not want to automate the grooming and nursing and caring, which they relate to dimming and switching the light at home.

With point of departure in these findings we propose a shift in discourse when talking about energy consumption in private households in relation to design of home automation, from that of functional engineering, to that of living, from a product focus on smart controls to a process focus on housekeeping, from conception of people as users to conception of people as co-producers of quality of life. This last point is inspired from Tim Ingold in 'Design and Anthropology' [25] where he states that designers should; "*look for a move from devising solutions that constrain practitioners to play by their own rules, to a position in which these rules are open to negotiation, and in which the improvisatory interventions of practitioners present an opportunity rather than a threat?"* [26: 32]. In the following section we expand on the implications of taking a housekeeping focus with regard to design.

3 A Design Discourse of Home-Keeping – What, Why and How

These findings, as presented in section 2, which we take as a designerly probing into users' living with the technology in question, and the collective resource approach, have led us to propose a strategy of design *with* users *for* digital living in the form of a home-keeping design discourse, for our case in point, communication about design of digital home control systems in the UserTEC project.

At first however we explain what we understand by 'design discourse'. Then we argue why a design discourse is a key to develop a design strategy in case of digital living with home automation.

A design discourse is the way in which designers – through their professional training – learn to explain to others what they do, what they can do, what their role in the complicated puzzle of actors building something is, and why. We find, that in order for designers to engage in design processes based on the collective resource approach, it is important, that they articulate what they, themselves bring to the table. In case of digital living, we as designers of home automation interfaces want to claim that we must (a) articulate user voices in ways which catch the attention of all stakeholders, (b) stage dialogues between all stakeholders, through which they see the value in their own context of taking user voices into account, and (c) argue convincingly for the business case of taking user voices into account from the very beginning.

Krippendorff [26] describes how design has moved from improving what is - Simon's definition of design in his seminal book the 'Design of the Artificial' [27] - to changing the meaning of what is. In an earlier account of a design paradigm of product semantics, Krippendorff characterizes the old design paradigm as one seeing design as making form follows required function, while the new he proposes, sees design as enabling users to make sense of things. In the old paradigm emphasis is on

simplicity and efficiency, while in the new emphasis is on self-evidence in identity and understandability – *"enables users to center themselves in a symbolically meaningful world"* [1:13]. This position moves the overall goal of design in the direction of designers seeing themselves as enablers of identity formation, enablers, who either make users' identity shrink, because they feel helpless, or grow, because they feel that they master the technology.

The fact that the two paradigms are confronted in today's design world, brings use and users, together with all other stakeholders, to the forefront, and creates, as Krippendorff notes, endless battles between technical oriented designers and user-centered designers. Krippendorf sees the outcome as a trajectory of focus shifts from utility, functionality and universal aesthetics, to marketability, symbolic diversity, and folk and local aesthetics, to natural interactivity, understandability and re-configurability, further on to informativeness, connectivity and accessibility, on to social viability, directionality and commitment. He claims that generativity, re-articulability and solidarity are on today's design agenda more than ever [27: 6].

We are inspired by Krippendorff's position that user-centered designers of today are motivated by challenges, opportunities to change something for the better, and possibilities to introduce variation, and that they must consider possible futures. Today designers' work is very much about formulating and altering discourses, an endeavor, which in itself forms a discourse, according to Krippendorff characterized by its artifacts and textural matter, its discourse community, its institutionalization of recurrent practices, its boundary, and its justifications [27: 32-37].

In Table 1 below we have systematized these characteristics of the user-centered semantic design discourse inspired by Krippendorff.

Table 1. Characteristics of a user centered design discourse inspired by K. Krippendorff

User-centered design discourse
Artifacts and textual matters: Research in user-centered design has in resent years moved towards the negotiation between stakeholders, inspired by the user driven innovation research [27][28] experimenting with the building of tangible models as communication tools for stakeholders
Discourse community: Instead of focusing solely on end user practices, more and more attention is given to the dialogue between mutual and conflicting interests of all stakeholders influencing the design outcome
Institutionalization of recurrent design practices: There is a growing awareness of the necessity to professionalize user-centered design, not as an ad on, but as discourse in its own right – in Krippendorff's words: *" … a concerted effort to improve the language and practice of design, its capacity to generate new proposals, to justify them to those who matter, their stakeholders, and above all, to make the redesign of design discourse a routine obligation"*[27: 35]
Boundary and justification: The boundary is expanded to include all stakeholders with a growing focus on dialogue between stakeholders and stakeholder discourses the justification of which being the ability of the user centered designers to formulate and reproduce user-centered design methods

However, in his approach, Krippendorff comes troubling close to marketing, and with his focus on design competencies he seems to cling to the idea of designing *for* users. And more importantly even, in our case, is the emphasis on language expressions. This means, that while we agree to turn away from 'changing what is' towards a joint explorations of 'what could be', we find the focus on language too limiting. Instead we build on Bakhtins position that meaning, to the extent that it can be shared, is shared in combinations of doing and talking, manifested in the collective resource approach in Scandinavian systems development in the joint effort of building prototypes, and also in the effort to involve all stakeholders in such activities, as has been developed in the research center SPIRE [27].

In Table 2 we modify Krippendorff's characterization of a user centered design discourse, in the spirit of the collective resource approach, not only in the interest of democratization as was the initial idea of the founding mothers and fathers, but also because todays quest for innovation makes listening to multiple contexts and allowing for self-regulated self-determining solutions mandatory.

Table 2. A collective resource approach to user-centered design

A collective resource approach to user-centered design
Artifacts and textual matters: Putting end-user practices and end-user innovation on the negotiation table in front of stakeholder-decision makers is the heart of collective resource approach to user-centered design.
Discourse community: The collective resource approach must include insights from the whole value network of a given design project, with openness to the possibility of finding new ways to solutions.
Institutionalization of recurrent design practices: At least in Scandinavia participatory design practices have been exercised over the last thirty years, and a canon of research almost always sited in participatory design research papers have developed, example being Schön's [29] work about the designer as a reflective practitioner. Also the building of tangible models as communication tools for stakeholders, incorporation of user-ideas and presentations of user practice has become commonplace in today's user-centered design practice.
Boundary and justification: The boundary circumscribe all stakeholders and requires that designers develop competencies in mediation and negotiation in local languages, but also that they take the responsibility to maintain focus in communication on end user practices

In the UserTEC project we will apply a collective resource approach to user-centered design, to guide our development of a communication strategy for keeping user practices in focus, and make it understood by all stakeholders in the project. This implies, that the democracy aspect becomes relevant in another way in the present context. Democracy research argue that for democracy taken in the political sense to live, it need a shared idea to fertilize the joint activities [27].

4 Home-Keeping and the Implications for Design

In the following we first elaborate on the choice of using the expression 'home-keeping', and then we argue why a collective resource approach in combination with a home-keeping discourse is a key to develop a design strategy in case of digital living with home automation.

The expression, presented in section 2; "*I find that it is very nice to stroll around and then push that one, 'now there will be light*" alerts us to a common human trait: the need to continuously, but in one's own pace and time, nurse and groom our dwelling, in order to feel truly at home.

Looking at this expression in a historical perspective reveals how having a dwelling, and maintaining it to be worth dwelling in, has been a key to survival, from ancient times to this day. We can sketch out a semantic net around the word 'dwelling' comprising location, being contained, safe, a point of departure/return, identity, sense of belonging, knowing enough, have choices, trophy-display - to name a few of the qualities of 'dwelling'. The most important point in relation to design work is that dwellings must be reproduced through continuous maintenance. It is this very act of maintenance, which produces the sense of containment, identity, and belonging, all emotional qualities, which new designs should try to match, should they become included in the everyday living in a household. We call this maintenance 'home-keeping', whereby we denote the intuitive drive to groom a place, make it your own, a place you build, rebuild, clean, and repair with what you have at hand, which by and large are important because of the maintenance of containment, identity, and belonging.

This is in line with the descriptions we find in Christopher Alexander's pattern language, pattern 79 "Your own home", where he quotes Martin Buber: *"…in the imperishable primal language of the human heart house means my house, your house, a man's own house. The house is the winning throw of the dice which man has wrested from the uncanniness of universe; it is his defense against the chaos that threatens to invade him. Therefore his deeper wish is that it be his own house, that he not have to share with anyone other than his own family. "* [31: 393]. We see here a close cobbling of house as home, and identity building and maintenance.

We take it, that one explanation why the inhabitants we talked to could live next to their IHC® control system without appropriating it, was that it did not prohibit them from appropriating their home in general, and that it did not spoil their sense of dwelling. If we are right, the important conclusion to draw is that, while home-automation is about doing away with routine tasks, and let the machine take over, home-keeping is about allowing inhabitants to interact with the machinery, groom with it, control with it, and develop it. These qualities of home-keeping can not be articulated only by the semantic turn suggested by Krippendorff. We must look for what is not already there, in language. We must first experience what the dwellers experience, and the formulation has to be a joint enterprise between all stakeholders. The difference between home-automation and home-keeping we see is parallel to what the design anthropologist Ingold describes in his work on lines [31], where he distinguishes between two well known ways of travelling: either we choose the most

efficient way of getting from a to b based on routine and prior knowledge, or we choose a causal adventurous modus, meandering our way forward, taking clues from the here-and-now environment, or in the words of the Norwegian anthropologist Fredrik Barth: we are looking for surprises [32].

We need to look for surprises because the solutions are yet unknown. Shared surprises are the source of a common ground for conceptualization of experience, as we referred to by quoting Bakhtin in the introduction: the main unit of meaning (not abstract sentences out of context) is formed through a speaker's relation to Otherness (other people, others' words and expressions, and the lived cultural world in time and place). Subjective by nature and definition, meaning, to the extent that it can be shared, is shared in combinations of doing and talking[10]. To operationalize this insight in design, we learn from the collective resource approach, as it has emerged out of the Scandinavian tradition in systems development.

5 Conclusion

In this paper we have discussed the problem of designing for digital home control, a prominent part of what we see as digital living, or living with digital technology.

Interviews with homeowners of IHC® show that users want to be in control of their environment, but if they do not see a way to take immediate action, they live with handing over the action to the system and having their power of control diminished. In other words the wish of the users is to be able to find *their* logic in the system; they do not want to automate the grooming and nursing and caring, which they relate to dimming and switching the light at home. This leads us to suggest a design discourse we call 'home-keeping'. In order to find a way to make user voices to influence a starting point for the design process we borrow the concept of design discourse mostly from Krippendorff's theory, combined with the Scandinavian tradition in systems development called the collective resource approach, and Bakhtin's theory of discourse as appropriation. The overall goal of a design strategy regarding home control devices, should then, as a minimum, not prevent inhabitants from appropriating their home, and ideally it should support the amalgamation of technology and living.

In more practical design terms user voices like the quotes presented here can be part of a toolbox supporting the development of a local shared vocabulary for design of home automation control displays, together with tangible prototypes, videos showing user practices, design game material, and workshop recipes. The toolbox – drawing on von Hippel's research in user driven innovation [33] facilitates a collective resource approach without requiring professional facilitators, hence it, in line with the toolbox developed in the DAIM project [28], can improve understanding between tech-companies and utilities and house-hold representatives about end user perspectives and enable end user perspectives to become drivers of innovation in smart technology development.

In conclusion we find Griet Scheldeman to be right, when she proposes that humans do not wish to glide effortlessly through life. That's what a lot of our technologies is offering, but she proposes that we might in fact need or like the

obstacles, play, bumps and effort to feel that we live: *"We should not just be concerned with how to reach the end but the means, practices and gestures as means to the end"* [26: 66].

Acknowledgments. We are grateful to the residents we visited, for hospitality and sincerity in their expressions, and to our reviewers for valuable input to the first draft of this paper.

References

[1] Krippendorff, K.: Product Semantics: A Triangulation and Four Design Theories. In: Väkevä, S. (ed.) Product Semantic 1989. University of Industrial Arts, Helsinki (1989), http://repository.upenn.edu/asc_papers/254 (retrieved)
[2] Gram-Hanssen, K.: Understanding change and continuity in residential energy consumption. Journal of Consumer Culture 11(1), 61–78 (2011)
[3] Gram-Hanssen, K.: Consuming technologies–developing routines. Journal of Cleaner Production 16(11), 1181–1189 (2008)
[4] Gram-Hanssen, K.: Residential heat comfort practices: understanding users. Building Research & Information 38(2), 175–186 (2010)
[5] Alexander, C.: Notes on the Synthesis of Form. Harvard University Press (1964)
[6] Grudin, J.: Computer-supported cooperative work: history and focus. Computer 27(5), 19–26 (1994)
[7] Bowers, J.: The work to make a network work, pp. 287–298 (1994)
[8] Nardi, B.A.: Context and Consciousness: Activity Theory and Human-computer Interaction. MIT Press (1996)
[9] Wertsch, J.V.: Mind as action. Oxford University Press, New York (1998)
[10] Bakhtin, M.M., Holquist, M., Emerson, C.: The dialogic imagination: four essays. University of Texas Press, Austin (1992)
[11] Spinuzzi, C.: Tracing Genres through Organizations: A Sociocultural Approach to Information Design. The MIT Press (2003)
[12] Bødker, S., Christiansen, E.: Poetry in motion: appropriation of the world of apps. In: Proceedings of the 30th European Conference on Cognitive Ergonomics, pp. 78–84. ACM, New York (2012), doi:10.1145/2448136.2448152
[13] Naur, P.: Programming as theory building. Microprocessing and Microprogramming 15(5), 253–261 (1985)
[14] Vygotsky, L.S.: Mind in society: The development of higher psychological processes. Harvard University Press, Cambridge (1978)
[15] Langefors, B.: Information systems theory. Information Systems 2(4), 207–219 (1977)
[16] Nygaard, K., Dahl, O.-J.: History of programming languages I. In: Wexelblat, R.L. (ed.), pp. 439–480. ACM, New York (1981)
[17] Göranzon, B., Fürth, T.: Datautvecklingens filosofi: tyst kunskap och ny teknik. Carlsson & Jönsson, Stockholm (1984)
[18] Josefson, I.: Språk och erfarenhet. Carlssons, Stockholm (1985)
[19] Ehn, P.: Work-Oriented Design of Computer Artifacts, Subsequent. Lawrence Erlbaum Assoc. Inc. (1990)
[20] Bjerknes, G., Bratteteig, T.: User participation and democracy: A discussion of Scandinavian research on system development. Scandinavian Journal of Information Systems 7, 73 (1995)

[21] Lytje, I.: Computer software as text. AI & Soc. 10(2), 142–163 (1996)
[22] Kanstrup, A.M., Christiansen, E.: Model Power – still an issue? In: Bertelsen, O.W., Bouvin, N.O., Krogh, P.G., Kyng, M. (eds.) Proceedings of the 4th Decennial Conference on Critical Computing: Between Sense and Sensibility, pp. 165–168. Association for Computing Machinery (2005)
[23] Wells Lynch, T.: The vision of a smart home has been kicking around for decades. Part two in our series about smart appliances. Reviewed.com (March 11, 2013)
[24] Leontiev, A.N.: Activity, consciousness, and personality. Prentice-Hall, Hillsdale (1978)
[25] Gunn, W., Donovan, J.: Design and Anthropology. Ashgate Publishing, Ltd. (2013)
[26] Krippendorff, K.: The semantic turn: a new foundation for design. CRC/Taylor & Francis, Boca Raton (2006)
[27] Buur, J., Matthews, B.: Participatory Innovation. International Journal of Innovation Management 12(03), 255–273 (2008)
[28] Binder, T., Brandt, E., Clarke, B., Halse, J.: Rehearsing the Future. Danish Design School Press, Copenhagen (2010)
[29] Schön, D.A.: The reflective practitioner: how professionals think in action. Basic Books (1983)
[30] Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press (1977)
[31] Ingold, T.: Lines: a brief history. Routledge, London (2007)
[32] Beach, D.: From fieldwork to theory and representation in ethnography. Studies in Educational Ethnography 11, 1–17 (2005)
[33] Von Hippel, E.: Democratizing innovation: The evolving phenomenon of user innovation. Journal für Betriebswirtschaft 55(1), 63–78 (2005)

The Digital Life of Vulnerable Users: Designing with Children, Patients, and Elderly

Alma Leora Culén and Maja van der Velden

University of Oslo, Department of Informatics, Gaustadalléen 23 B
Oslo, Norway
{almira,majava}@ifi.uio.no

Abstract. Vulnerability is about being at risk and it is often understood as the effect of limited physical or cognitive capabilities, such as age, frailty or illness. Vulnerable people are frequently excluded from the design of technologies that could in fact support them in tackling these risks. This paper explores designing with three vulnerable groups: children with special needs, chronically ill teenage patients, and isolated, or afraid of being so in the near future, elderly adults. We choose three distinct groups in order to show the breadth and variations in the ways in which people may be vulnerable. We looked at their digital lives and possible new risks and dependencies created by the use of digital technologies. Designing with vulnerable people is practically, methodologically, and ethically challenging. We show how methodological and reflexive sensibilities help to address these challenges and keep the design process on track.

Keywords: Vulnerable users, interfaces, teenage patients, children with special needs, elderly, design of privacy, reflexivity, social networks.

1 Introduction

Information and communication technology (ICT) is no longer limited to desktop computers. It is ubiquitous and pervasive, permeating everyday objects and activities. The increasing availability and use of digital technologies in our daily lives – at home, in our neighborhoods, at work, in school, etc. – suggests a new form of digital living. Living the digital life brings new opportunities, possibilities, and challenges. It also brings new dependencies, such as the fear of losing access. In the case of the elderly, we see also the fear for having to use these technologies [1].

Not everyone is equally served by the opportunities and possibilities of digital technologies. In the 1990s, the term the digital divide was used to describe the people(s) and countries excluded from access to digital technologies, in particular the Internet. In the early years of the 21st century, a more sophisticated view on the digital divide developed. It was not simply about 'haves' and 'have nots', but about social inclusion and effective use [2, 3].

In HCI, a similar development is described. The first wave of HCI focused on better human-machine fit, on usability. The second wave, originating in cognitive

M. Aanestad and T. Bratteteig (Eds.): SCIS 2013, LNBIP 156, pp. 53–71, 2013.

science, emphasized theories of what is going on not only within a computer, but also in a human mind. HCI is currently in its third wave with its focus on experience design and situated design [4]. There is more talk about socio-materiality, phenomenology, design thinking, dialogue, values, social issues etc. and much less talk about the design-as-engineering approach of the earlier waves of HCI.

Similarly, interface design has moved from design for the average user [5] towards recognition that there is no such thing as an average user. This line of thinking has continued and culminated in "design for all" [6] and universal design [7]. Such technology design for all strives to design for human diversity, social inclusion, and equality.

There are people and groups of people who are still excluded from the digital life. Digital exclusion is often described as related to age, gender, and/or socio-economic status. In this paper, we take a closer look at vulnerable users and their use of digital technology. In the literature, vulnerable users are often described as people at risk because of their age, frailty, diagnosis or limited capacities, both physically and cognitively. In our view, the term 'vulnerable user' is an inclusive term. All of us may become vulnerable users at one time in situations such as illness, temporary disability, or inability to deal with some new technology in our environment.

Vulnerable user groups provide additional challenges for designers, due to the lack of appropriate design methods, difficulty in communication or the difficulty to empathize with vulnerable users' experience of the world [8], given their often reduced motor skills or cognitive and/or social abilities.

In this paper, we consider three groups of vulnerable users: young learners with developmental, cognitive, physical or emotional impairments, chronically ill teenage patients, and elderly adults living alone. The aim of the paper is to discuss how technology or interface design processes with and for vulnerable users can look like in practice. We present a design case involving each of the selected vulnerable user groups. We discuss 1) some of the challenges in uncovering the needs of these user groups; 2) challenges in applying known methods for designing with users, in particular participatory design; 3) challenges in modifying known methods or finding new ones that work better for these specific groups; and 4) some of the ethical challenges involved in working with vulnerable users.

Other researchers and research communities have been concerned with vulnerable users. Prior work included research into homeless persons, elderly suffering from dementia, children with special needs, and others. For example, [9] considers children to be vulnerable online and classifies them as vulnerable due to economic, social, psychological, or physical factors. However, much of the prior work does not consider vulnerability through the lens of designing technology with these users. One noteworthy and recent exception is the CHI 2013 workshop on "Designing with and for Vulnerable People", see [10], in which papers covered different aspects of addressing vulnerability through design were presented. The workshop papers addressed vulnerable groups such as homeless, isolated elderly, bereaved, demented elderly, vulnerable women in secure hospital settings, children with special needs, people with low health literacy and others at risk socially, physically, cognitively or economically. The workshop's focus was similar to that of this paper (we participated with a position paper on children with special needs), bringing forth challenges of designing in sensitive contexts with people at risk. In spite of challenges, studies with

vulnerable users are much needed. In particular the development of methodologies, addressing ethical challenges, and ways of evaluating effectiveness of design solutions on vulnerable users' lives.

The structure of the paper is as follows: the next section describes designing for vulnerable children, illustrated by a learning app design for a special education class in a primary school. Section three describes a case of designing social media privacy settings with teenage hospital patients. This section is followed by design for elderly, where design efforts aim to reduce the sense of loneliness and increase the understanding of how elderly view technology in their lives at present. In section five, we discuss the methodological and ethical challenges we experienced in our studies, including parallels and differences between our groups, followed by some concluding remarks.

2 Designing with and for Young Learners with Special Needs

Today's children are often called the 'millennials', digital natives and the like. However, many millions of children and students between the age of 6 and 17 have a condition that impairs their ability to participate in a typical classroom environment [11]. Many researchers have considered new technologies and their impact on disabilities, e.g. [12]–[14]. Pervasive, mobile classroom technologies have been integrated into the classroom ecology [15]. Papers describing the use of the tablets in education, and how they help children with special needs, appeared both in scholarly venues [16]–[18] and mass media [19]. Literature on designing applications especially for and with these children is still scarce, although there are some examples, most frequently of children with ADHD and autism [20].

Designing technology with children [21] is always a challenging task. Designing for and with children who have special needs is particularly challenging [22, 23]. For these children, the combination of power relationships, often reduced communication skills, and additional stakeholders such as parents, teachers or caretakers, results in a very complex situation. Participatory design, seen from the Scandinavian perspective, implies involving the children as equal partners in the design process. In the case of vulnerable children, being equal may be difficult. Thus, they risk being excluded from participation in the design of technologies that will be used by them. If they are included in participatory activities, high ethical standards are needed to protect children and to prevent negative consequences [24, 25]. Designing, even adapting a piece of technology to a student's need [16], is challenging, as techniques may need to change in order to accommodate for the particularities of student's needs. The work is more time consuming and may require other specialists as part of the team. Design solutions are rarely cost-effective because they often need customization and are used by a small number of children. At the same time, this user group may benefit greatly from new educational assistive technologies.

2.1 Research with Young Learners with Special Needs

The part of the study described in this paper took place from January – June of 2012. The iPad was considered as a cool piece of technology, and Norwegian schools were

just starting to adopt it as a learning tool [26]. An elementary school was interested in assessing the iPad as a tool to stimulate learning in their special education class.

The group we worked with consisted of a small class of 6 boys aged 8-12, a teacher, and two assistant teachers, one of whom was dedicated to one of students with extra needs. Each student had received an iPad equipped with the same applications as the rest of the children at school. However, none of the apps could be used for teaching as our group of children had different abilities and competence levels. The teacher was interested in finding out if an app could be designed that would fit all the students in the class.

We had prior experience with app design for a whole class [27] in a regular classroom. On our first visit to the class, we engaged the children in simple and fun iPad games, observed what they did with the iPad on their own, and talked to their teachers. All the children could use the iPad as a device: they could open apps, regulate volume, size of images and text, etc. On the other hand, problems related to understanding of what a selected app does or actions available to them were immediately apparent and prevented the children from using the apps according to their goals. This visit was followed by a long interview with the main teacher about specific issues and problems for each student, as well as how the class functions as a whole. The main finding from this interview was that in spite of the fact that the boys have been going to the same class for years; there was little feeling of being a group. Information about cognitive, social and physical development levels of each student, to the level pertinent to our project, was given to us. We categorized the issues we heard about and found out that they are comparable to those in Livingstone [9, p. 22]. The students had education/economic disadvantage, psychological disadvantages, disabilities (some of them) and social disadvantages. Further, they exhibited behavioral problems, impulse control problems, social problems and, according to the teacher, most relevantly, concentration problems.

2.2 Methodological Challenges

We usually use participatory design (PD) methods in designing with and for children. As mentioned above, the teacher told us that the children in this special class did not work or communicate well together. That was also what we saw during the initial session with the children. Thus, the usual teamwork involving two to four children was replaced by working with one child at a time. Furthermore, each child presented a unique design challenge. One could see that our overall goal of designing one app for the whole group was going to be much more challenging than our previous work [27].

In order to find out if the iPad could be used as a learning tool for the class, we organized two workshops aimed at gaining insight in how the iPad and game based learning function as motivators for learning mathematics or improving language skills, see Fig. 1. The workshops were divided into ca 30 minutes sessions. Each session involved one child, one of the authors, the teacher and a graduate student. In other situations when designing with children, we match the number of adults with the number of children so that the children do not feel even less power due to the sheer number of adults present. In this situation, the boys were very comfortable with

their teacher, and we have done our best to establish good contact during the initial visit. Still, the sessions involved three adults and a single child. In an attempt to compensate a bit for this fact, the sessions were made as playful as possible. Children's eagerness to work with the iPad and curiosity as to what kind of games they get to play was helpful as well. We were interested in the amount of time a child could concentrate on a game without interruptions. We were also interested in having the children complete at least one cycle of the game, even if it required extra time and assistance.

Fig. 1. A child with strongly reduced sight is trying a spelling app. He normally uses special assistive technology for people with reduced vision. The student says that the iPad is way cooler than his equipment. All apps tested with children were from the Apple app store.

The analysis of data collected from these workshops showed large diversity in abilities among the children. Some had cognitive difficulties and could not read or write, some could not do any math, and others could not memorize musical tones. We were challenged by this diversity. Additionally, it was clear from what we observed during workshops that the children could not represent themselves and their needs well, if at all. The teacher was acting on children's behalf, by both stating what their needs are and interpreting their actions and behaviour.

All the children and the teacher were very positive during the workshops. We could identify learning opportunities for every child, but failed to do so for the class as a whole. Encouraged by the teacher, we made a decision to shift towards mastery of some daily task rather than learning math or language. Inspired by [28], we decided to engage more actively with values in the design process, and focus less on the method itself, PD and representation of stakeholders.

Consequently, a new workshop was organised, with the same duration and participant organization as earlier. This time, we used physical objects (lots of coins of different denominations, beans etc.) and role-played purchasing scenarios. Some iPad apps with shopping games were tested as well, followed by a short interview session where the children were asked about their experiences with physical vs. digital shopping and game preferences. This did give us some ideas about the types of games they like to play as well as visual style preferences. We obtained some insights that were helpful in further design. The challenge was how to organize all these activities so that a child remains an active participant, given concentration issues that were common for all. We found that, even though we were starting out with role-play, games or questions in the same way with each child, the sessions were different. Some children needed these activities to be interlaced, while others would carry them out one after the other. Some level of improvisation was needed with every child in order to keep them interested and focused. Occasionally it was difficult to interpret a child's behavior, e.g. it was difficult to correctly identify the source of excitement: was skipping between apps an expression of excitement and desire to show what they can do, or a reflection of concentration difficulties? We also found out that, if this was a project larger than an app design, other experts e.g. occupational therapist or a special pedagogue, should be included as part of the design team.

Interviewing the children was also interesting. As Teachman and Gibson conclude in [29], the quality of data gathered through interviews is always depending on the interviewer and a good toolkit. In our case, the time before the children would lose interest and concentration was short. In the case of vulnerable children, we felt that we needed a stronger, perhaps more versatile toolkit than usual. The challenges in co-designing with vulnerable children are in part due to communication difficulties and difficulties in interpreting children's behavior correctly. For example, when a child exited the room under a conversation, we were unsure if the child found the interview to be boring, or was acting on an impulse. Thus, the sessions required additional awareness, sensitivity, willingness to improvise and adjust techniques, find new ones, invest more time and possibly other resources.

After the third workshop with children, the final concept for the app was developed, and the prototyping phase started. The app and the design are further described in [30]. The initial design phase presented here illustrates the issues and challenges for researchers when working with this vulnerable user group.

3 Designing Privacy with Teenage Patients

Chronically ill teens (12-18 years old) have a long-lasting or persistent health condition, which requires continuous treatment, often for the rest of their lives. From developmental and disease management perspectives, chronically ill teenagers are considered a distinct group of patients. The teenage period is viewed as preparation for the transition to the adult-centred healthcare system; they take increasingly more responsibility for their health and treatment [31, 32]. At the same time, this period is characterised with decreasing compliance with treatments and therapy [33]–[35]. Teens may have the impression that their therapy or treatment may make them feel different from their peers or even feel defective [36]. They may experience that they are treated as children while they want to be treated as young adults [37, 38].

In our research with teenagers, we focus on how interactive technologies are, and can be, used to support their information and communication needs as patients. We are especially interested in their online privacy management. One of the aims of our study is to contribute to the development of methods for including teenage patients in the design of technologies that matter to them, and that alleviate some of the risks associated with their vulnerabilities. In this paper, we report on the ethical and methodological challenges of working with teenage hospital patients as participants in design for privacy settings that will enable them to distinguish clearly who gets to know what about them. Apart from being socially vulnerable, these patients also may suffer from physical or psychological vulnerabilities. In addition to looking at designing privacy for this group, we also look at the physical body positioning and limitations on the use of technology while, for example, being in horizontal position during long periods of time.

Including young patients in the design of interactive technologies is an import requirement for patient centered design and participatory design [39]–[41]. Designing with young patients is perceived as difficult because of the extra challenge involving ethical issues and consent [42]. Secondly, they may have self-esteem issues or not associate their personal identity with their diagnosis, and may therefore not want to participate on the basis of being a patient [43]–[45].

3.1 Research with Hospitalized Teens

There is a large body of work on teenage users of social media and privacy, but none of it addresses young patients. We therefore implemented in 2011 a study among 20 teenage patients in a children's hospital in Canada [43]. The analysis of the qualitative interviews showed, among other things, that all teens were active on Facebook and implemented a wide range of privacy strategies. Facebook was, however, not used to share or discuss their diagnosis. It was important for these teens to keep their patient identity separate from the social identities they present and explore online. On the other hand, several teens expressed their desire to meet other teens with the same diagnosis in order to exchange experiences and to support each other.

In a follow-up study, in 2012, 16 teens participated while receiving treatment in the hospital. We implemented a new version of the 2011 study, this time based on a card sorting exercise. This study confirmed our earlier findings on teenage patients and online privacy. They also participated in the re-design of the visibility of Facebook's privacy settings, using paper prototyping, and were asked to describe their ideal patient social network. As a result of the findings in these two studies we began to use the 'cool wall' [46, 47] in interviews with Norwegian teens, non-patients and non-hospitalized patients. The latest iteration of the 'cool wall' is now being used in design research with teenage patients in a Norwegian children's hospital.

Teenage patients receiving treatment in a hospital are often isolated from other patients and are immobile because of their treatment or as an effect of their condition. Our contact time with the patients had to be organized between doctor visits, treatments, family visits, visits to physiotherapy, hospital school attendance, and homework. It was almost impossible to have non-interrupted time with the

participant. Thus, the context of the study imposed some limitations. Secondly, the majority of the participants were lying in bed, frequently with one arm connected to hospital equipment, thus limited in their movements and in what they could accomplish during a session. In addition, hospital equipment, e.g. a dialysis machine, affected the quality of the audio recordings during the interviews and design work.

Another practical issue, with consequences for our choice of design methods, was that materials used in the workshops had to use the limited space available for display. They had to fit a hospital tray table and they had to be mobile – so that all the props could easily be picked up and moved whenever necessary. The design process itself needed to be flexible – so that the discussion was easily picked up after an interruption; and privacy aware – so that private information didn't get disclosed accidentally to third parties. Also, the choice of materials was affected: everything that was shared between the patients needed to be disinfected. The cards and other paper props were therefore plasticized.

3.2 Methodological Challenges

During the second study in the Canadian children's hospital we used two methods: card sorting and paper prototyping. In Norway, we are using a third method, the 'cool wall'. We discuss each approach in turn.

3.2.1 Card Sorting with Thinking Aloud

The card sorting technique was used to find out what teenage patients think about online privacy. They were asked what they share on Facebook and with whom. We created a set of 52 plasticized cards, based on the interviews with teenage patients in the first study [43]. We asked the participants to associate the word on the card with Facebook and to sort the cards over three piles: positive, negative, and neutral. While they sorted the cards, we asked them to think out loud. We then asked them to select the top 3 positive and negative associations. Almost all participants were eager to explain the reasons for sorting cards in a particular way. Of the 16 participants in the card sorting exercise, two were not able to sort the cards themselves. One participant had to lie flat and could therefore not use her bedside table to sort the cards. One participant could not use either arm. Some movements in the card sorting game easily triggered the alarm from the intravenous system. This would prompt a visit from the medical stuff, leading to an interruption in the process.

3.2.2 Prototyping

The prototyping exercise was based on a set of plasticized cards with design alternatives for the visibility of privacy settings in a social network. We then worked on the personalization of one alternative design, using paper, markers, and a pen on the hospital tray table. The example in the prototyping exercise was based on Facebook. All participants were experienced Facebook users and therefore felt familiar with the task at hand. In order to work on the paper prototype, the participant had to be able to sit up fairly straight in bed in order to have a good view on the paper. This was uncomfortable for some patients and impossible for the participant who was lying flat. In these cases, we asked the participants to tell us how they would do it.

The researcher would implement it and make sure this was the way the participant had in mind. We felt that this way of working was not particularly inspiring for the participants. The quality of their prototyping was lower than in the case of participants doing it themselves.

3.2.3 'Cool Wall'

Looking for other and maybe more appropriate ways to include teenage hospital patients in design work, we were inspired by the research on 'cool' and the 'Cool Wall' [46, 47] (see Fig. 2 for an earlier iteration of the 'cool wall' made out of a metal panel, magnets, sticky tacks, and paper icons (see Fig. 2), The 'cool wall' is used to find out which social media applications are considered cool and what are cool things to do on a patient-centered social network. The concept of 'cool' is of large importance to this age group, and for understanding of what matters to teens in general, and this patient group in particular. 'Cool' may be a determining factor in use patterns of any piece of technology or any interface designed for teenage use.

Fig. 2. A 'cool wall' inspired by [47]. The figure shows the first iteration of the design, where focus is on being able to detect what about social networks is considered to be "cool". The latest iteration can be mounted on a flexible arm, using magnets that are strong enough when the wall is hanging over the patient's bed.

This section illustrates how the usual tools for uncovering patients needs such as paper prototyping are of limited use to patients, in particular if they are in bed. We have

further shown how the cool wall technique was adapted to just such situated use. The challenges for us as researchers were strongest when it came to ethical issues and the question of ongoing consent. This issue is further discussed in section five. As the teenage patient's vulnerabilities most often are unrelated to cognitive or communication problems, methodological challenges were related to adaptation of tools and techniques to their situation and environment. Both card sorting and prototyping examples point out the difficulties in applying these methods, as well as the need to carry out these techniques with materials that are easy to sterilize. The cool wall example, re-designed for teenage patients, was on the other hand a great tool for understanding young patient's privacy needs and it served as a starting point for an excellent master thesis [48] on a closed social networking site for teenage patients.

4 Designing with Elderly Living Alone

In contrast to children and teenage patients, the digital natives, elderly adults are often not comfortable with new technology, even when they are healthy. This is especially true of low education and low-income older elderly (age group over 80) [49]. At the same time, one should be careful not to assume that all shy away from technology or are unable to learn how to use it. Diversity among elderly in the use of technology, as well as motivation to adopt it when offered, is large. They are often depending on technology to extend their stay in own homes eventhough their abilities deteriorate with ageing. Gero-technology is a research area that refers to the design and use of technologies that promote independence and autonomy, enabling elderly to live longer in their own homes, as well as providing the support networks for social inclusion [1]. One of the largest gero-technology projects in Europe is the Ambient Assisted Living (AAL), focusing on enabling active living and social support for Europe's aging population. According to the Norwegian Ministry of Health [50], three main causes of moving elderly from their own homes to health care units are: falls, cognitive decline and loneliness. The elderly just starting to experience these problems can now opt to live in Care Plus homes, where active living is encouraged, and living units have some smart-house technologies such as light and heat sensors and touch screens [51] for communication and entertainment. In addition, Care Plus homes often offer a smart gym [52], for collective or individual training, as well as other social and cultural activities.

Gradual decline in visual and auditory perceptions, motor skills and cognitive abilities make elderly into a highly non-homogeneous user group in terms of physical and cognitive abilities. This affects their motivation, ability and self-efficacy in using ICT. This has implications for technology and interface design [53]–[57]. No solution fits all. Multimodal interactions and interfaces are an emerging field and include, in addition to touch interfaces and mouse/keyboard, interactions with a system through voice [58], gestures [59], tangible objects or the whole body interactions. Technology design processes used to include participants who are less vulnerable and still active, if they engaged elderly at all. Recently, there is an effort in the CHI research community to design with intended users [60, 61].

4.1 Research with Elderly Living Alone

We narrow our focus on design and use of communication technologies by elderly living alone. Even when still vital and active, elderly often suffer from fear of social isolation and loneliness. As mentioned above, loneliness is one of the three main reasons for elderly living alone to move to a care unit. In [62], the authors conclude that loneliness is a predictor of functional decline and death. Loneliness is defined as the subjective feeling of isolation, not belonging, or lacking companionship. Living alone does not necessarily imply loneliness. It is fully possible to live alone and not feel lonely, as well as it is possible to be surrounded by others and still feel lonely. However, we have chosen participants for our study among those living alone since they are at higher risk of feeling lonely the older they get. In [63], the authors show that living alone predicted lower psychological well-being and loneliness worsened the effects of living alone. In this sense, elderly living alone are considered to be vulnerable users.

During the fall semester of 2012, two student-based design projects [64, 65] were carried out in two suburban elderly centers. Both projects focused on the sense of loneliness among the elderly. The issue came in focus in part from the centers' leadership, and in part from the literature and the elderly themselves. The students have discovered that elderly living in the centers had low technological literacy and both projects ended with design and implementation of a solution similar to Skype, but with larger icons and in general, simpler and better suited interface for this user group.

Seeking to get a richer picture of both communication needs and technologies used to carry them out, we reconsidered the approach to the problem and decided to look at somewhat younger, still active elderly aged 67 and up to 80, living alone in own homes. The age of 67 is the usual retirement age in Norway. Our aim was to explore the issue of loneliness and the role and possibilities existing, and future technology has to alleviate the risk of feeling alone. This research is ongoing. In this paper, we present a tool we developed, the communication wall, along with some initial findings.

4.1.1. 'The Communication Wall'

Inspired by social mapping, and in particular [66], we have developed a communication wall to help us explore communication patterns of elderly and technologies they use. We considered their social lives, service needs, communication patterns at present, issues with these, desires for other things and finally, free fantasy about future products and technologies.

The tool kit for the Wall consist of three maps, colored pens, eraser, and many icons representing different communication possibilities, both platforms and communication channels: Skype, email, mobile phones, smart phones, tablets, desktop computers, chat, social media, land-line phones, as well as images of future technologies such as holo-phones displaying holographic images of a person one talks to, chips embedded under the skin on the inside of an arm, with dialing possibilities etc. The participants also had an option of drawing anything else they could think of and include in the Wall. The first map, 'Friends and family communication', shows the people they are close to, as well as means of communication with them. Participants are instructed to place the people who are physically closest to them in

the first circle, people in the same country in the second, and then those in the rest of the world. Alternatively, they could arrange the contacts by frequency of communication. The second map, contacts and services needed, shows the services they use, such as health services, travel services, libraries, cultural places, restaurants etc. and technology they use to contact the service providers. The last map, the 'Future technology', shows the same information, but participants are asked to place the future technologies if any are seen as desirable. The constructed maps and patterns were an excellent starting point for discussion about future technologies, while the process itself enabled discussion on the subject of loneliness and the sense of connectivity to others, with and without technology.

Fig. 3. An elderly woman, age 73, is using the iPad to take the photo of her own Wall. On the right, top row shows a Wall of a 75 year old women, while the bottom row shows the Wall of a 67 year old man.

4.2 Methodological Challenges

When working with healthy and active elderly, it is common to use focus groups or workshops. However, we wanted a tool that would enable the collection of a much richer and perhaps more intimate data than it is possible to do in a group setting. Thus, the communication wall was developed for working on the one-to-one basis with participants. While working with a participant, we opened a space for talking about isolation, loneliness, ethical problems in this kind of research and dialogue on present and future technology use in every session. The communication wall has facilitated this conversation in a very nice way. As a participant considered and talked about those people that are close to him/her, for some participants, very personal and powerful memories from the past and/or concerns about the future were brought forth.

The challenge in the aftermath of the data collection process is to find appropriate analytic tools that would enable some of these reflections to be kept through discussion of findings rather than losing them in categorizations and coding. Furthermore, the intended use of this data was to enable us to better understand patterns of use and needs of active elderly in order to better address needs of elderly in care units and elderly centers. Thus, the active elderly were a kind of proxy users for those who are older, lonely and in many cases, have reluctance towards, sometimes even fear of technology. However, we were not necessarily interested in simple solutions such as proposed in [64, 65], but rather solutions that reflect values of elderly and extend beyond voice based communication. Some of the data collected may be used for that purpose, but further work with the communication wall and users that are closer to this user group is needed.

5 Discussion and Concluding Remarks

The three cases represent some of the design work we have initiated with people whose age, frailty, diagnosis or limited capacities, both physical and cognitive, often excluded them from participation in technology design processes. Involving these vulnerable users as designers in the design of their digital lives requires, however, considerable adaptation of existing design methods as well as the development of new methods, including better methods of working with proxy user groups, as in the case of elderly.

In all three cases, we experienced that the participants were eager to participate in our study. The technology designs - learning games with the school children, privacy settings with teenage patients, and video conferencing and the communication wall sessions with elderly adults living alone – were perceived as relevant to their lives.

The first challenge, and one of the major we experienced, was related to finding out about the technology needs of vulnerable users. In the case of the school children, these needs are often formulated by other stakeholders, in particular the teacher and the school management. Teenage patients were able to express their own technology needs. The elderly adults living alone in nursing homes couldn't express their technology needs. Their lack of familiarity with technology hinders them to verbalize what they might want or need. On the other hand, the younger, active elderly living alone were highly eloquent in talking about technology and even thinking about their future needs. Thus, in some ways they do and in some ways they do not adequately represent the group living in care units. How to bridge this gap is a challenge we need to address through future research.

It became clear, in all three cases, that designing with vulnerable users often needs to take place on a one-to-one basis. We could not create a proper participatory design process with the three groups of users. Each school kid had such particular challenges, that we had to relinquish the idea of designing one learning application with the whole group of children. The hospitalized teens were separated from each other because they were confined to their individual rooms – some of them in isolation. The active elderly needed the private space in order to get an opportunity to look into their very private feelings of loneliness and communication patterns. The number of connections shown on their communication walls could not be taken as indicators of their well-being, and

we could see that clearly through working with them. The communication wall was an enabler for expressing the feelings around the topic of loneliness.

Another methodological challenge was the cognitive and physical ability of the participants. Attention span is an issue in designing with children, but becomes especially challenging when designing with children with special needs as a result of ADHD and other cognitive disabilities. This was also an issue with some of the teenage patients because of the medication they were receiving, which often made them drowsy. Other physical challenges that affected the design work included the inability to use their arm/hand or to sit up, in the case of the teenage patients and tremors and callous fingers in the case of the elderly.

Ethical challenges are a major issue when working with vulnerable users. The most important one concerns consent. This consent needs to be informed and ongoing. Our consent forms provided information about the study, what kind of information was collected, and how the information gathered was going to be used. The consent form also stressed the issue of voluntary participation.

The research with young patients was particularly challenging. While the hospital's Research Ethical Board took care of the procedural part of the consent process, by approving the text of the consent form, the issue of ongoing consent remained unaddressed. How to confirm consent and voluntary participation during the course of the interview and design work? The issue of power plays an important role in a situation in which an adult interviews or works with a child. This issue gets an extra dimension when the child, in our case a teenage patient, is lying in bed and can't walk away from the situation or may find it difficult to express being tired.

Reflexivity, being aware of one's position and role as an adult and researcher and continuously reflecting on this role in the meetings with the teenage patients, became our main approach to this challenge, e.g. [57, 67]. We tried to be attentive to changes, such as changes in the patient's position in bed, the patient's voice and facial expressions, as well as changes in the treatment. A change could result in questions such as: Are you not too tired? Do you want to take a break? Also, a change in the research set-up, which created a transition, presented a good opportunity to re-confirm consent.

Design research with vulnerable users requires a large amount of flexibility and improvisation. Our own experiences and capacities as researchers were often challenged, as we needed to adapt to new situations for which there was no standard method or approach. Instead of a leading a participatory design process, we began to follow the participants: What do they enjoy doing? What is possible? What do their answers or activities tell us about how to proceed? At the same time, we realized that we needed to develop this methodological sensibility [68, 69] into a more reflexive sensibility [69]: Why do we interpret the participants' use and design practices in this particular way and not that way? These sensibilities helped us to keep us focused on the aim of our research, namely how to design *with* vulnerable users for vulnerable users, not solely designing for vulnerable users.

The three cases also show that a high level of ethical reflexivity is required. Unequal power relations, ongoing consent, and voluntary participation needed our constant attention. In particular because we met our participants in locations which they were not expected to leave, e.g. at school, in a hospital, or in their own home.

We had to confirm regularly if our presence was still appreciated and their participation was still consented. This was particularly difficult in the case with the school children with special needs. We were often not able to explain the impulsive behavior of some of the kids: did they tell us that they were no longer interested in participating or was this part of their usual behavior?

Designing with vulnerable users didn't only challenge us methodologically and ethically. Our emotional responses to these experiences, and the physical environments in which they took place (special education classroom, hospital, care homes), brought out our own vulnerability. We were sometimes unsure if our responses to unanticipated situations were the right ones. In some cases, we shared the fact that we needed to improvise with our participants. This opened up for dialogue, which we experienced as positive and supportive and not necessarily a weakening of the position of the researcher or of the design process as a whole.

5.1 Concluding Remarks

In this paper, we conceptualized vulnerable users as particular groups of people who, because of their physical or cognitive abilities, are also not able to make their voice heard in the design of their digital lives. We presented participatory design work with three groups of vulnerable users and described some of the methodological and ethical challenges we experienced. The particular design contexts of our study required us to adapt our participatory design approach. In order to make sure that we would continue to 'hear the participants' voices' – to design with them, not only for them – we needed to develop a reflexive sensibility. This sensibility focused our design efforts as well as broadened our interpretative frame when we were confronted with situations and behavior that we found difficult to explain.

Acknowledgments. We would like to thank our graduate students, Anna Karpova and Margaret Machniak, as well as the students from the Kamfer and Selskap projects for their contributions to this study. We would like to thank all the participants for their interest, time, and participation. We also thank the Children's Hospital of Eastern Ontario (CHEO) and Khaled El Emam of the Electronic Health Information Laboratory (Ottawa) for their support. A component of the study is implemented as part of the research project Autonomy and Automation in an Information Society for All, which is funded by the Verdikt Program of the Norwegian Research Council (project number 193172).

References

1. Künemund, H., Tanschus, N.: Gero-Technology: Old Age in the Electronic Jungle. In: Komp, K., Aartsen, M. (eds.) Old Age In Europe, pp. 97–112. Springer, Netherlands (2013)
2. Warschauer, M.: Reconceptualizing the digital divide. First Monday 7(7) (2002), http://firstmonday.org/ojs/index.php/fm/article/view/967

3. Gurstein, M.: Effective use: A community informatics strategy beyond the Digital Divide. First Monday 8(12) (2003), http://firstmonday.org/ojs/index.php/fm/article/view/1107
4. Bødker, S.: When second wave HCI meets third wave challenges. In: Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles, pp. 1–8. ACM (2006)
5. Newell, A.F.: Extra-ordinary human-computer interaction. In: Edwards, A.D.N. (ed.) Extra-Ordinary Human-Computer Interaction: Interfaces for Users With Disabilities. CUP Archive (1995)
6. Stephanidis, C. (ed.): Universal Access in HCI, Part I, HCII 2011. LNCS, vol. 6765. Springer, Heidelberg (2011)
7. Goldsmith, S.: Universal Design. Routledge (2012)
8. Grudin, J.: Groupware and social dynamics: eight challenges for developers. Commun. ACM 37(1), 92–105 (1994)
9. Livingstone, S., Haddon, L., Görzig, A., Ólafsson, K.: EU kids online II: final report (2011), http://www.eukidsonline.net/
10. Designing for and with Vulnerable People, CHI 2013 workshop (2013), http://di.ncl.ac.uk/vulnerability
11. Hasselbring, T.S., Glaser, C.H.W.: Use of computer technology to help students with special needs. The Future of Children, 102–122 (2000)
12. Alper, M., Hourcade, J.P., Gilutz, S.: Interactive technologies for children with special needs. In: Proceedings of the 11th International Conference on Interaction Design and Children, pp. 363–366. ACM (2012)
13. Wise, P.H.: Emerging Technologies and Their Impact on Disability. The Future of Children 22(1), 169–191 (2012)
14. Mintz, J., Branch, C., March, C., Lerman, S.: Key factors mediating the use of a mobile technology tool designed to develop social and life skills in children with Autistic Spectrum Disorders. Computers & Education 58(1), 53–62 (2012)
15. Merchant, G.: Mobile practices in everyday life: Popular digital technologies and schooling revisited. British Journal of Educational Technology 43(5), 770–782 (2012)
16. Gasparini, A., Culén, A.: Tablet PCs – An Assistive Technology for Students with Reading Difficulties? In: The Fifth International Conference on Advances in Computer-Human Interactions, pp. 28–34 (2012)
17. Hourcade, J.P., Bullock-Rest, N.E., Hansen, T.E.: Multitouch tablet applications and activities to enhance the social skills of children with autism spectrum disorders. Personal Ubiquitous Comput. 16(2), 157–168 (2012)
18. McClanahan, B., Williams, K., Kennedy, E., Tate, S.: A Breakthrough for Josh: How Use of an iPad Facilitated Reading Improvement. TechTrends 56(3), 20–28 (2012)
19. iPad gives voice to kids with autism. CNN, http://www.cnn.com/2012/05/14/tech/gaming-gadgets/ipad-autism/index.html
20. Benton, L., Johnson, H., Ashwin, E., Brosnan, M., Grawemeyer, B.: Developing IDEAS, pp. 2599–2608. ACM Press (2012)
21. Druin, A.: Mobile Technology for Children: Designing for Interaction and Learning. Morgan Kaufmann (2009)
22. Frauenberger, C., Good, J., Keay-Bright, W.E.: Designing Technology for Children with Special Needs - Bridging Perspectives through Participatory Design. CoDesign: International Journal of CoCreation in Design and the Arts 7, 1–28 (2011)

23. Guha, M.L., Druin, A., Fails, J.A.: Designing with and for children with special needs: an inclusionary model. In: Proceedings of the 7th International Conference on Interaction Design and Children, pp. 61–64 (2008)
24. Lewis, A., Porter, J.: Interviewing children and young people with learning disabilities: guidelines for researchers and multi-professional practice. British Journal of Learning Disabilities 32(4), 191–197 (2004)
25. Ljusberg, A.L., Brodin, J., Lindstrand, P.: Ethical issues when interviewing children in remedial classes. International Journal of Rehabilitation Research 30(3), 203–207 (2007)
26. Culén, A., Gasparini, A.: Situated Techno-Cools: factors that contribute to making technology cool. PsycNology Journal 10(2), 117–139 (2012)
27. Culén, A., Gasparini, A.: When is a Student-centered, Technology Supported Learning a Success? International Journal of Digital Information and Wireless Communications 2(3), 256–269 (2012)
28. Iversen, O.S., Halskov, K., Leong, T.W.: Rekindling values in participatory design. In: PDC 2010, pp. 91–100. ACM, New York (2010)
29. Teachman, G., Gibson, B.E.: Children and youth with disabilities: innovative methods for single qualitative interviews. Qualitative Health Research 23(2), 264–274 (2013)
30. Karpova, A., Culén, A.: Challenges in designing an app for a special education class. In: IADIS, Interfaces and Human Computer Interaction 2013, Prague, July 22-24 (2013)
31. Alpay, H.: Transition of the Adolescent Patient to the Adult Clinic. Peritoneal Dialysis International 29(suppl. 2), 180–182 (2009)
32. Callahan, S.T., Winitzer, R.F., Keenan, P.: Transition from pediatric to adult-oriented health care: a challenge for patients with chronic disease. Current Opinion in Pediatrics 13(4), 310 (2001)
33. Kyngäs, H.: Compliance of adolescents with chronic disease. Journal of Clinical Nursing 9(4), 549–556 (2000)
34. Taddeo, D., Egedy, M., Frappier, J.-Y.: Adherence to treatment in adolescents. Paediatr. Child Health 13(1), 19–24 (2008)
35. Nevins, T.E.: Non-compliance and its management in teenagers. Pediatric Transplantation 6(6), 475–479 (2002)
36. Friedman, I.M., Litt, I.F.: Adolescents' compliance with therapeutic regimens: Psychological and social aspects and intervention. Journal of Adolescent Health Care 8(1), 52–67 (1987)
37. van Staa, A., Jedeloo, S., van der Stege, H.: 'What we want': chronically ill adolescents' preferences and priorities for improving health care. Patient Preference and Adherence 5, 291–305 (2011)
38. Lugasi, T., Achille, M., Stevenson, M.: Patients' Perspective on Factors that Facilitate Transition from Child-centered to Adult-centered Health Care: A Theory Integrated Metasummary of Quantitative and Qualitative Studies. Journal of Adolescent Health 48(5), 429–440 (2011)
39. Waller, A., Franklin, V., Pagliari, C., Greene, S.: Participatory design of a text message scheduling system to support young people with diabetes. Health Informatics Journal 12(4), 304–318 (2006)
40. Berwick, D.M.: What 'Patient-Centered' Should Mean: Confessions of an Extremist. Health Affairs 28(4), 555–565 (2009)
41. Reis, C.I., Freire, C.S., Fernández, J., Monguet, J.M.: Patient Centered Design: Challenges and Lessons Learned from Working with Health Professionals and Schizophrenic Patients in e-Therapy Contexts. In: Cruz-Cunha, M.M., Varajão, J., Powell, P., Martinho, R. (eds.) CENTERIS 2011, Part III. CCIS, vol. 221, pp. 1–10. Springer, Heidelberg (2011)

42. Lang, A.R., Martin, J.L., Sharples, S., Crowe, J.A., Murphy, E.: Not a minor problem: involving adolescents in medical device design research. Theoretical Issues in Ergonomics Science, 12 (2012)
43. van der Velden, M., Emam, K.E.: "Not all my friends need to know": a qualitative study of teenage patients, privacy, and social media. J. Am. Med. Inform. Assoc., 16–24 (2013)
44. Chira, P., Nugent, L., Miller, K., et al.: Living Profiles: Design of a health media platform for teens with special healthcare needs. Journal of Biomedical Informatics 43(5(suppl.)), S.9–S.12 (2010)
45. Mack, R., Giarelli, E., Bernhardt, B.A.: The Adolescent Research Participant: Strategies for Productive and Ethical Interviewing. Journal of Pediatric Nursing 24(6), 448–457 (2009)
46. Fitton, D., Horton, M., Read, J.C., Little, L., Toth, N.: Climbing the cool wall: exploring teenage preferences of cool. In: Human Factors in Computing Systems Extended Abstracts, pp. 2093–2098. ACM (2012)
47. Read, J., Fitton, D., Cowan, B., Beale, R., Guo, Y., Horton, M.: Understanding and designing cool technologies for teenagers. In: Human Factors in Computing Systems, pp. 1567–1572. ACM (2011)
48. Machniak, M.: Sprinklr: Designing a 'cool' cealth-oriented social networking site with and for teenagers. Master Thesis, University of Oslo (2013)
49. Leist, A., Ferring, D.: Technology and Aging: Inhibiting and Facilitating Factors in ICT Use. In: Wichert, R., Van Laerhoven, K., Gelissen, J. (eds.) AmI 2011 Workshops. CCIS, vol. 277, pp. 166–169. Springer, Heidelberg (2012)
50. Helsedirektoratet: Welfare technology. Report on implementation of welfare technologies in the municipal health and care services 2013-2030 (2012)
51. Culén, A.L., Bratteteig, T.: Touch-Screens and Elderly users: A Perfect Match? In: The Sixth International Conference on Advances in Computer-Human Interactions, pp. 460–465 (2013)
52. Culén, A., Finken, S.: Design and Interaction in a Smart Gym: Cognitive and Bodily Mastering. In: SouthCHI 2013, July 1-3 (2013)
53. Dickinson, A., Arnott, J., Prior, S.: Methods for HCI research with older people. Behavior & Information Technology 26(4), 343–352 (2007)
54. Howthorn, D.: Possible implications of aging for interface designers. Interacting with Computers 12, 507–528 (2000)
55. Fisk, A.D., et al.: Designing for older Adults: Principles and Creative Human Factors Approaches. CRC Press, Boca Raton (2009)
56. Werner, F., Werner, K., Oberzaucher, J.: Tablets for Seniors – An Evaluation of a Current Model (iPad). In: Wichert, R., Eberhardt, B. (eds.) Ambient Assisted Living. Advanced Technologies and Societal Change, vol. 2, pp. 177–184. Springer, Heidelberg (2012)
57. Warin, J.: Ethical Mindfulness and Reflexivity Managing a Research Relationship With Children and Young People in a 14-Year Qualitative Longitudinal Research (QLR) Study. Qualitative Inquiry 17(9), 805–814 (2011)
58. Portet, F., Vacher, M., Golanski, C., Roux, C., Meillon, B.: Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects. Pers. Ubiquit. Comput. 17, 127–144 (2013)
59. Vatavu, R.-D.: Nomadic gestures: A technique for reusing gesture commands for frequent ambient interactions. Journal of Ambient Intelligence and Smart Environments 4, 79–93 (2012)

60. Vines, J., Clarke, R., Wright, P., McCarthy, J., Olivier, P.: Configuring participation: on how we involve people in design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 429–438. ACM, New York (2013)
61. Lindsay, S., Jackson, D., Schofield, G., Olivier, P.: Engaging older people using participatory design. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, CHI 2012, pp. 1199–1208. ACM, New York (2012)
62. Perissinotto, C.M., Stijacic C., I., Covinsky, K.E.: Loneliness in older persons: a predictor of functional decline and death. Archives of Internal Medicine 172(14), 1078–1083 (2012)
63. Lim, L.L., Kua, E.-H.: Living Alone, Loneliness, and Psychological Well-Being of Older Persons in Singapore. Current Gerontology and Geriatrics Research 2011 (2011)
64. Åmdal, V.S., Klette, K., Simonsen, V.S., Ahmedov, S.: Student project "Selskap", http://www.uio.no/studier/emner/matnat/ifi/INF2260/h12/projects/elderly/Elderly-aid-app/
65. Voje, A., Eggen, H., Gabrielsen, T., Vitikka, T., Bech, L.: Student project "Kamfer", http://www.uio.no/studier/emner/matnat/ifi/INF2260/h12/projects/elderly/Kamfer/
66. Living Profiles–Map your Social Tree, http://livingprofiles.net/?page_id=99
67. Phelan, S.K., Kinsella, E.A.: Picture This... Safety, Dignity, and Voice—Ethical Research With Children Practical Considerations for the Reflexive Researcher. Qualitative Inquiry 19(2), 81–90 (2013)
68. Mörtberg, C., Studedahl, D.: Silences and sensibilities: increasing participation in IT design. In: Proceedings of the 4th Decennial Conference on Critical Computing: Between Sense and Sensibility, pp. 141–144 (2005)
69. Law, J.: After Method: Mess in Social Science Research. Routledge (2004)

Software Innovation –Values for a Methodology

Ivan Aaen

Department of Computer Science, Aalborg University, Denmark
ivan@cs.aau.dk
www.essence.dk

Abstract. Innovation is a recurrent theme in public as well as academic debate, and software development plays a major role for innovation in about every sector of our economy. As a consequence, software innovation will play an increasingly important role in software development. The focus in this paper is on how to make innovation more likely to happen in software development at the level of the software team or project. At this level it is important to identify opportunities to create added value in ongoing projects. Changes in software technologies over the last decades have opened up for experimentation, learning, and flexibility in software projects, but how can this change be used to facilitate software innovation? This paper proposes a set of values to guide the development of a methodology to facilitate software innovation.

Keywords: Software innovation, creativity, software development, Agile Manifesto, values, Hegelian dialectic.

1 Introduction

Just about everybody is for innovation, everybody talks about it and finds it vitally important, but there is a shortage of methodological advice on how to make innovation more likely in software development – at least at the level of the team or project.

Software innovation represents a class of problems including the development of innovative software products, designing software support for innovative business processes, transforming known solutions to innovative uses in new contexts, and stimulating paradigmatic changes among developers and customers concerning the framing of use context and the discovery of potential game changers on the market.

Neither traditional nor agile software development methods offer much advice on software innovation. The classic challenge for software development is to combine quality and efficiency; and software engineering – whether traditional or agile – is concerned with delivering quality solutions in a predictable and effective way. But today's world is different. We need more than just meeting requirements effectively to stay competitive; we need to create high value solutions.

In light of globalization and the increasing commoditization of IT providing a shared and standardized infrastructure [9], we need to move from mainly operational considerations towards more strategic ones: Software development in high-cost countries must achieve more valuable results than overseas software development [3], and

M. Aanestad and T. Bratteteig (Eds.): SCIS 2013, LNBIP 156, pp. 72–86, 2013.

we should therefore look for principles for software development beyond the traditional efficiency and quality focus: How can software teams deliver high value solutions? This is the topic of this paper.

The paper starts out by outlining software innovation as a concept and surveys contributions within the field (Section 2). Section 3 suggests four values for software innovation based on a discussion of traditional and agile development from an innovation perspective. Section 4 applies these values to a simple example to illustrate how these values may drive innovation in software development. Section 5 discusses implications of these values for a methodology for team-based software innovation.

2 Software Innovation – Concepts and Contributions

Software innovation is not really established as a term yet, and existing contributions to this emerging field are scattered over several organizational levels and stages of the development process. Some contributions are generic and have little focus on either organizational level or particular stages [33]; some have a strong focus on the company level [30]; some on picking and improving promising project proposals [19]; some on ideation in the requirements stage [24, 25]; and some on innovation as part of an ongoing project [1, 2, 8, 12].

This paper is aimed at the methodology level for software teams. It is part of building a foundation to help software teams increase the value of what they build as they go about building it. There are numerous techniques and tools for creativity and many insights on how to stimulate creative thinking and innovative work, but very little work has been done on methodology for software teams.

Hirschheim et al. define information systems methodology as *an organized collection of concepts, methods, beliefs, values and normative principles supported by material resources* [22]. As software innovation is close to information systems development, we use this definition as inspiration for developing a software innovation methodology. Such a methodology must stand on a set of values, and in this paper we will discuss and suggest one such set of values.

We focus on innovation as part of the software development project. We see software innovation as part of everyday life in a team, and thereby as part of what the designer and in fact any stakeholder engaged in a software development project does. We focus less on what happens before a project is decided and more on what takes place from the decision to start a project until the end of it. The aim is to find ways to help a team develop, mature, and implement ideas as part of a development project. We assume that modern development techniques will be used to allow for iteration and experimentation within reasonable levels of risk. In other words, we see software innovation as a learning process where experiences and insights during a project may change its course.

We understand software innovation as concerned with introducing innovation into the development of *software intensive systems* as defined in the international standard ISO/IEC/IEEE 42010, i.e. systems in which software development and/or integration are dominant considerations. Our focus is on innovations that offer something new to

known users or customers, or something known to new users or customers. Specifically, we do not include changes in the software development process itself into our understanding of software innovation. Software innovation here refers to the user or customer side only.

Innovation usually extends creativity in the sense that ideas are developed and matured in the context of implementation. Basically ideation is concerned with the generation of socially acceptable ideas [37], whereas innovation by definition implies change in the real world [40].

Whereas the interest in software innovation is fairly recent, there has been some interest in ideation and creativity since the early and mid 90ies. J. Daniel Couger worked on creative problem solving [14] and creativity techniques [15, 16] for information systems development. More recently, Ben Shneiderman worked on creativity support in the same field [36]. Within software engineering, Neil Maiden has worked on creativity workshops [26] and stakeholder collaboration [25].

Contributions with a direct bearing on software innovation are still relatively few. Within information systems development, innovation research tends to focus on the business context and adoption of innovations. For example Burton Swanson [39] advocate mindfulness when innovating with IT. Within software development, there is some interest in innovation as a goal for software development. Jim Highsmith and Alistair Cockburn point to the potential for agile development to support innovation [21], but as Conboy et al. [13] observe based on a number of studies on the relationship between agility and innovation or improvisation: *These have tended to focus more on the agile practices themselves as the innovation and not the extent to which the practices facilitate agility and innovation*.

In the last few years a growing number of writers have published very varied work on software innovation.

Jeremy Rose [33] proposes eight work-style heuristics for software developers, and Pikkarainen et al [30] offer eight fundamental practice areas for innovation with software, each containing a number of activities at the company level to master that particular practice.

Misra [28] presents a goal-driven measurement framework for software innovation processes linking these metrics to business goals. The processes per se are not part of this framework. Also at the business level, Gorschek [19] suggests Star Search, an innovation process using face-to-face screening and idea refinement for software-intensive product development. This process has particular focus on ideation and selection prior to actual development.

Focusing on the team level, Aaen discuss how to facilitate software innovation [1] and suggests using roles in innovative software teams [2]. At a similar level, Conboy and Morgan [12] discuss the applicability and implications of open innovation in agile environments using two examples from industry, and Mahaux and Maiden [24] suggest using improvisational theater as part of requirements elicitation to support team-based innovation. The main focus is on improving stakeholder communication, increasing mutual understanding, and generating ideas that can be expressed as requirements.

These very varied contributions generally focus on methods and normative principles for software innovation, whereas concepts, beliefs, and values still seem to be missing in this field. Consequently there seems to be no methodological framework available for developers and other stakeholders facing the challenges in software innovation. Proposing a set of values is therefore one step in building such a methodological framework.

3 New Values for Old: Bridging Two Paradigms

In 2001 the Agile Manifesto [4] presented a critique of the traditional paradigm for software development. The manifesto marks a milestone in software development and reflects important developments in software technologies. It was indeed a new paradigm and it has impacted strongly on software practices around the world.

The manifesto held promise for innovation by changing focus back to software itself more than elaborate requirements and processes. The four values and twelve principles in the manifesto promoted iterative, agile, and evolutionary development and marked a fundamental departure from traditional software engineering [32]. This change opened up for experimentation, learning, and flexibility in software projects.

The manifesto emerged from remarkable developments in software technologies and developer competencies after the traditional software engineering paradigm was established in 1968 [29]. Developments such as patterns, refactoring, automated testing, object orientation, software libraries, IDEs, self-organizing teams, and more all support flexibility and reduce overheads related to change. These developments allow for alternative ways to work effectively and achieve quality results in software projects.

The clash between the two paradigms was succinctly presented in the manifesto by expressing agile values over traditional ones:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*

Although these traditional values were never stated in this exact form, they nevertheless express core ideas in traditional software engineering curricula. Overall, both paradigms aim for efficiency and quality in software development and the conflict expressed in the manifesto is largely about *how* to develop software. The purpose of software development itself – *why* to develop and *what* to achieve – is not addressed much in either paradigm. These questions, however, become increasingly urgent when focus change from production to innovation. Software innovation implies that software development must aim for more than efficient delivery of quality solutions to known requirements.

In this section we will try to move beyond the two paradigms in search for values to stimulate and steer software innovation. We will focus on what unites the two, and based on this unity focus on innovation. This discussion will be inspired by Hegelian dialectics.

Hegel (1770-1831) saw knowledge as constantly changing; coming about via a perpetual struggle between contradictions – tensions and paradoxes. Any initial *thesis* (position), when analyzed, has confusions, flaws, or deficiencies, which in time will lead to the formulation of an *antithesis* (negation). This antithesis in substantial ways contradicts the original thesis. The confrontation between the thesis and antithesis leads to new tensions, and from these tensions a *synthesis* (negation of the negation) is formulated in an attempt to resolve the original contradiction while preserving and maintaining insights from this contradiction into a new understanding.

Essentially, Hegelian dialectics asserts that knowledge must pass through a phase of negation to arrive at a synthesis where useful portions of an idea are preserved while moving beyond the limitations of it.

The following discussion will move from the struggle between two production paradigms towards a synthesis to support software innovation.

3.1 New Value: Reflection over Requirements

The Agile Manifesto values *Customer collaboration over contract negotiation*. This value is about customer requirements: How do we know what to deliver?

Traditional development focuses on creating stable and known conditions to make it possible to deliver on time, within budget, and according to requirements. Contracts therefore are based on requirements specifications, and these specifications in turn are used to verify quality [31].

Agile development expects needs to change with circumstances [6]. Therefore development is incremental and requirements are determined and prioritized at the beginning of an iteration to ensure relevance at this moment in time.

Despite their differences, the traditional and agile approaches are in full concert on purpose: To comply with customer requirements. Compliance basically is about doing what one is asked to do. To software development projects, this implies ideas to be essentially external to the development process - more so in the traditional than in the agile world. Both approaches are about conforming to customer requirements effectively, irrespective of whether they are detailed in specifications or handed over by a customer representative. Both approaches strive to answer the same challenge: How can software teams live up to customer requirements?

Software innovation must answer a different challenge: How can software teams deliver high-value solutions? This challenge obviously includes meeting requirements, but goes beyond this traditional goal. Innovation is about learning; about discovering what could or should be done; about exploring new technological possibilities and application options. Innovation is about reflecting on needs and discovering opportunities, and by that token about exceeding customer expectations via joint exploration.

Therefore, software innovation replaces the principle of customer collaboration over contract negotiation with a new value: *Reflection over requirements.* Reflection combines customer insights from the application domain with the technological expertise of the developer to excel customer expectations. Reflection corresponds to a process where working on a solution cannot be separated from developing a deeper

understanding of the problem itself and its use context. For this reason, we should not become prisoners of previously stated requirements, but continuously reflect on their relevance.

3.2 New Value: Affordance over Solution

The Agile Manifesto values *Working software over comprehensive documentation*. This value is about what is produced in a project: Which artifacts are important?

Traditional development use elaborate models to ensure precision in the product, to make products maintanable, and to be able to settle disputes in case the outcome is questioned [31].

Agile development is based on a simple observation: Such models are usually not what customers want [6]. Most customers value software over documentation, and they want solutions to meet current needs. Partnering contracts, incremental delivery of useful solutions, and recurring testing in realistic settings all serve to ensure that customers get what they need most without spending too much on things less wanted.

Both paradigms understand development as essentially about delivering solutions requested by the customer. In this sense, project deliverables essentially are what developers are asked to deliver, i.e. solutions to stated needs.

In software innovation the relation between problem and solution is often complex. Most designs have the potential to offer more, and as we work on the problem in the use context, we discover untapped potential in our solutions that we may choose to exploit [27].

What a given design offers – its *affordance* – therefore comes into focus. Gibson [18] defines the affordance of a thing as a specific combination of properties with reference to a particular being. Affordance is what a thing can be used for more than what it was designed for. A software designer may therefore consider what a design at any given moment can afford to the user beyond what was required precisely to solve those particular needs that caused the design to come up in the first place. In other words, any design might afford features beyond what was anticipated at the time of design. Such features invite to discover new perspectives in the use context.

Innovation that synthesizes application domain potential with technological potential is not a natural part of either paradigm. Therefore, software innovation replaces the principle of working software over comprehensive documentation with a new value: *Affordance over solution*. The response to a given challenge for a software project may neither be derived from known requirements and models nor from user stories. Instead, the solution may emerge as the project develops and options and potentials are discovered through a continued dialogue between application domain needs and potentials afforded by technology.

3.3 New Value: Vision over Assignments

The Agile Manifesto values *Responding to change over following a plan.* This value is about adaptation: How can a project adapt to changing business needs?

Traditional development value comprehensive planning. Detailed plans facilitate resource allocation and task assigments [31], but obviously plan-based projects experience problems when conditions are dynamic.

Agile development embrace change via continuous customer interaction and iterative development to suit customer needs as they change. This is why agile projects employ adaptive or rolling wave planning [23]. Allocation of resources and tasks are taken care of by self-organizing teams.

The two paradigms use different ways to plan and assign tasks to team members. Traditional project management is formal and explicit, while agile project management is informal and based on personal commitment. Still, both approaches are concerned with task assignment derived from customer requirements.

Assignments will always be handed out, one way or other. They are given, and thereby the person receiving the assignment is largely excluded from the innovation process; the innovation process is essentially over at the time of assignment. If we want to allow for innovation throughout the project, we must redefine project management to one of developing and maturing a vision of the project. Innovative project management must refine both scope and goals of the project as it unfolds, and the vision must inspire the team and stimulate creativity [38].

Therefore, software innovation replaces the principle of responding to change over following a plan with a new value: *Vision over assignments.* If we want to pursue a fleeting target while still being able to know where we are heading and work together towards a shared goal, we must replace the traditional requirements-based plan with a project vision. As we learn from working on problems and solutions, our vision at a given time – what American pragmatist John Dewey called the *end-in-vie*w [17] – is transformed itself by our work. We therefore need a representation that can be shared and easily revised.

3.4 New Value: Facilitation over Structuration

The Agile Manifesto values *Individuals and interactions over processes and tools.* This value is about team organization and work processes: How can work processes support a team?

Traditional development focus on processes based on standardization, control, and metrics, where the software process tends to become an object in itself. The process is detailed in business manuals on corporate webpages, and the process structures the overall project as well as what individual team members do [10, 11].

Agile development processes are communities of practice [43], where individuals and teams develop their processes incrementally via interactions and personal competencies. Processes may be just as stable as is the case in traditional development, but they are developed through daily practices and constantly adapted to the current situation. Such processes are often referred to as empirical processes [23].

Both approaches focus on structuration - on stabilizing work processes either via externalization in defined processes, or via learning in communities of practice. They both focus on stability and repeatability and are therefore essentially conservative.

In software innovation, the software process must facilitate innovation. It must offer flexible support for learning and for developing solutions, and our focus must change from the process itself and how to make it stable and repeatable, to supporting improvisation and reflection under fleeting conditions. The process must support and promote lateral thinking, discovery processes, open up for building new paradigms, growing unconventional ideas, and so on.

Therefore, software innovation replaces the principle of individuals and interactions over processes and tools with a new value: *Facilitation over structuration.* This facilitation includes techniques for creativity as well as for evaluating and maturing ideas and visions.

4 An Illustration

We will use a very simplified example to illustrate how these values could drive software innovation. The example is inspired by an ongoing project and concerns a system to support rehabilitative physiotherapy for newly operated patients. We assume the patients to do part of their rehabilitation training at home.

Could we design a system to help patients and therapists collaborate virtually? A system where patients exercise in their home under therapist supervision using an Internet connection?

We will use this example to illustrate how a software development team could develop innovative solutions combining technological expertise with application domain insight. This development is described in four prototypes, where each prototype is described from four points of view. Each view in turn is related to one of the values described above. The four views furthermore represent four fundamental concerns in software projects [7].

The four views are:

1. *Paradigm* – representing the problem from a user/customer perspective. The view is called paradigm because it reflects underlying mental models of the application domain, including who the users are and what the market is or wants. This view represents the *use context* of the system: An understanding of the domain the system is meant be part of. This understanding reflects users and needs combined with options and it may change in the course of the project.
2. *Product* – representing how solutions could be implemented. The product view sees the product from the 'inside' with a focus on architecture and possible ways to build features. This view is used for designing the system *configuration*, the platforms and components used to build features for the use context. The hardware platforms may or may not change much between prototypes, whereas the features built on these platforms can change considerably over time.
3. *Project* – representing plans, status, and priorities in the project. This view is for project management. In lieu of a requirements specification, project management here is based on a project vision to set the course for the project and

represent a shared goal. For the sake of brevity we will use simple metaphors [5, 44] to represent visions here.

4. *Process* – offering a range of tools for idea generation and indeed idea evaluation. This view is used for process facilitation to help generate ideas in the project via creativity techniques, or improve and mature ideas via evaluations [42]. Using creativity techniques is straightforward so we will focus on evaluations here. Evaluations aim at identifying potentials in simple ideas or more comprehensive visions, and at supporting decision-making in the project.

Using these four views, the following four sections describe the development of prototypes, where one prototype inspires the next. This development is driven by the four values for software innovation described above. The parallel development in four views are shown in Fig. 1. Driven by the values proposed above, the prototypes change as we learn about the use context and about what is afforded by the platforms used for building the product.

1st Prototype: X-ray

The first reflection on the use context could be one where the system is used for real-time interaction with therapists giving *instructions* on how to do exercises to IT-savvy patients in their home. Scenarios for this could include:

- Repertoire management: Register the repertoire of exercises for the patient.
- Exercise instruction: Instruct and monitor the patient doing a given exercise.
- Therapist's log: Exercise history, patient repertoire, status and progress.

The prototype could consist of a Microsoft Kinect camera in combination with a video link, a laptop in the patient's home, and a cam-equipped PC in the therapist's office. The Kinect would compensate for some of the limitations of the video-feed by highlighting bones, joints, and movements of the patient. This would help the therapist see how the exercise is performed and give instructions to the patient. Similarly, the patient would be able to see the therapist demonstrate the exercise and ask questions for clarification.

The metaphor describing the initial vision for this prototype could be *X-ray*. This vision reflects the initial motive for using a Kinect: To compensate for limitations in patient-therapist interaction based on a video feed. The Kinect is used to enhance those parts of the feed that are most important for a therapist in order to assess how an exercise is performed.

An evaluation of the qualities of this version could facilitate the development of the next. In our example, an evaluation could identify as a strength, that this prototype would indeed allow the therapist to instruct patients. The system serves as a simple medium between the therapist and the patient, and a weakness therefore is that the contribution to the users is limited. The present design is threatened by poor economy due to the limited benefits, and vulnerable to competition due to low entry barriers. On the other hand, the system may afford more due to unused potential in the technical platform.

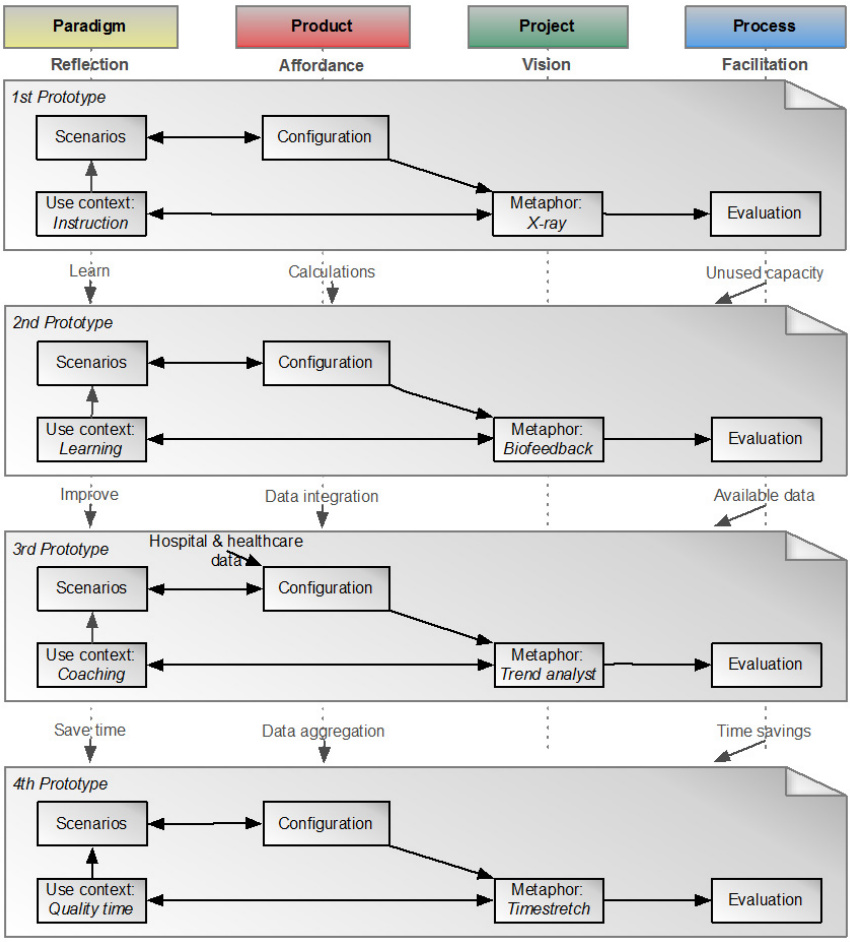

Fig. 1. Prototype development as seen from four viewpoints

2nd Prototype: Biofeedback

The system could monitor the patient's exercises and compare in real time with a model of how these exercises are supposed to be performed. Reflecting the addition of such features, the use context for the second prototype could shift to one of *learning*.

Information on the patient's screen might indicate whether a performed exercise match with prescription. This way, the patient may improve via self-observation as a supplement to therapist monitoring. Scenarios added for the modified use context could include visualizations of a performed exercise versus the prescribed version, quality indicators for the exercise, indicators on moments in an exercise where improvement is needed, etc.

Changing the use context to one of learning calls for more reflection on the patient side of the system. How can the patient receive better feedback and help on how to do

the exercises correctly? The second prototype could employ augmented reality on the patient side, for example by synthesizing skeleton points on screen as they would move in a perfect exercise and show the patient's actual movements for comparison. Alternatively, the patient's skeleton points could change colors and/or have arrows added if movements deviate too much from prescriptions. A third option could be to synthesize a colored region on the screen to indicate the zone of a correct exercise. Skeleton points within the region would indicate a correct exercise. These are all examples of how to provide the patient with immediate feedback to stimulate the learning process.

The vision for the second prototype could change to *Biofeedback,* adding more benefits to the patient side of the system without sacrificing the therapist side. This vision sees the patient as empowered and engaged. The system provides information to the patient on the relevant parts of the motoric system under rehabilitation.

Facilitating further development, an evaluation of this prototype could identify as a strength that the patient works more independently with this system and may require less support from the therapist. A weakness could be that there is quite limited support and benefits for the therapist. There are still quite low entry barriers and competitors may therefore threaten the system. There seems to be untapped potential in integrating data from the patient side with data from the hospital side and healthcare in general; affording such a combination could allow the therapist to give better advice to the patient.

3rd Prototype: Trend analyst

Having added features to detect the quality of exercises performed by the patient and thus also the quantity of them, it would seem relevant to reflect on the therapist role again and change the use context to *coaching*. By aggregating data from the patient into a rehabilitation history it would be possible to analyze progress over time. Such records could help the therapist decide on when to adjust an exercise – for example adding more weight – or when to change to other exercises.

Features added in this prototype would combine data from the patient side with historical and standard data on the therapist side. These data could support predicting the performance in current session and comparing these predictions with actual data to see if the patient trains too much or too little between sessions, or if problems with e.g. scar tissue from surgery seem to impair rehabilitation. Such features might help prevent overtraining syndrome and setbacks, and possibly also predict the duration of the rehabilitation process.

This prototype has *trend analyst* as vision to support the therapist as a coach for the patient. A coach has a longer time perspective and looks for improvements and problems between sessions.

To facilitate further development, an evaluation of this prototype could help identify options and affordances. A strength of this prototype could be that the combination of patient data and hospital and healthcare data will increase functionality and heighten entry barriers to protect against competition. A weakness could be that the system mainly gives quality improvements. A threat to the system could therefore be

economy as productivity gains are small. An opportunity might be to use available computer power on the patient side to aggregate exercise data and visualize these in condensed form. This could afford servicing more patients.

4th Prototype: Timestretch

Having changed the use context to focus more on the therapist and added features for analyzing exercises as performed by a patient, we may look for affordances to improve therapist performance. The therapist is engaged in timesharing – moving from one patient to the next, monitoring performance, and communicating with the patient. The monitoring of a patient prior to coaching takes time if performed in real-time, but the aggregated data makes it possible to compress the time spent on monitoring. This could allow the therapist to start coaching sooner. As a reflection of this, the use context could focus more on *quality time* with the patient, spending less time on preparations. From the patient's perspective, this solution could give automated feedback from the system in-between therapist interactions, and help the patient identify issues to discuss with the therapist.

Scenarios for this context could include visualization of recent exercises aggregated on the screen, aggregation of quality indicators, problem indicators, fatigue warnings, patient history, and more.

Seeing the use context as one of quality time necessitates focusing less on trivialities and more on essentials in interactions. Features in this prototype would contribute to higher productivity and stronger support for the patient. Apart from features supporting the use scenarios described above, new features for the prototype could include wizards to help the patient stand in a correct position vis-à-vis the Kinect camera to ensure quality input to the system. Other features could advice the patient on likely topics for an upcoming talk with the therapist. Such advice could come from indicator data derived on the therapist side of the system.

The *Timestretch* vision suggests productivity gains. If the therapist can coach a patient effectively and use less time doing so, there are possibilities for either improving rehabilitation effects for a patient, or for rehabilitating more patients with the same number of therapist hours.

Decisions on whether to use the features afforded by the fourth prototype is facilitated by another evaluation. A strength of this prototype is that it adds productivity gains to all the benefits offered in the previous prototypes. A weakness might be an increased risk of occupational stress, as the therapist will be able to service more patients. An opportunity could be to design data aggregation carefully to offer easy-to-use indicators for the therapist to use when working with a patient. A threat might be that the system is less useful for atypical patients, where standard data and indicators might not apply.

5 Discussion

This paper suggests four values for software innovation and illustrates their implications in a small example. The values are thought as part of a methodology for software innovation at the level of team or project.

Working on prototypes using these four values and their respective viewpoints supports the exploration of use context and helps identify options and challenges. The use of prototypes for experimentation and exploration is similar to Donald Schön's ideas on designing as a reflective conversation with the materials of a situation [34]. Such conversations often involve dialogues across fields of expertise. As we build them, we may discover features on the platforms that allow for richer solutions, or discover available data that could improve the quality and scope of a solution. Looking opportunistically for affordances [41] in and around the configuration may add value to the solution.

The four visions are really extensions from the first to the last. The features defined as part of the X-ray metaphor are useful in any of the later prototypes. This fortunate situation may be common, but obviously a change in project vision will sometimes be disruptive and perhaps costly. As the vision develops in a project, it will usually grow increasingly stable as the product is tested with customers and users. Still, it remains changeable and allows the project to adapt to changes by guiding without excessive detail [20].

Using evaluations is an important way to improve the design of a solution. Indeed, feedback can serve as a stimulus to creativity in software design [27]. Evaluations and judgments may be according to predefined or ad hoc criteria, and they may even be tacit and intuitive [35]. In software innovation we use evaluations and other process elements to facilitate idea development as we work on solutions while reflecting on the problems we try to solve.

The prototypes discussed here illustrate some practical implications of the values. In incremental development projects and in particular in agile projects organized in sprints or similar, the deliverables after each sprint may serve as prototypes. Such a prototype corresponds to a perceived use context and a project vision, and it consists of a configuration of features and platforms. This can be evaluated in connection with a sprint review meeting, and context, vision, and configuration may change if the team chooses to search for more valuable solutions.

The values presented in this paper serve as a basis for an ongoing effort to develop a methodology for software innovation called *Essence*. These values drive the development of views, roles, vision representations, evaluations, use of affordances, etc., that form part of Essence.

References

1. Aaen, I.: Essence: Facilitating Software Innovation. European Journal of Information Systems 17, 543–553 (2008)
2. Aaen, I.: Roles in Innovative Software Teams: A Design Experiment. In: Pries-Heje, J., Venable, J., Bunker, D., Russo, N.L., DeGross, J.I. (eds.) IS Design Science Research. IFIP AICT, vol. 318, pp. 73–88. Springer, Heidelberg (2010)
3. Aspray, W., Mayadas, F., Vardi, M.Y.: Globalization and Offshoring of Software. Association for Computing Machinery, Job Migration Task Force (2006)

4. 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., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development (2001)
5. Beck, K.: Extreme Programming Explained: Embrace change. Addison-Wesley, Reading (2000)
6. Beck, K., Andres, C.: Extreme programming explained: embrace change. Addison-Wesley, Boston (2005)
7. Bernstein, L., Yuhas, C.M.: People, Process, Product, Project - The Big Four. In: Bernstein, L., Yuhas, C.M. (eds.) Trustworthy Systems Through Quantitative Software Engineering, pp. 39–71. John Wiley & Sons, Inc., New York (2005)
8. Campos, P.: Promoting innovation in agile methods: two case studies in interactive installation's development. International Journal of Agile and Extreme Software Development 1, 38 (2012)
9. Carr, N.G.: IT Doesn't Matter. Harvard Business Review 81, 41–49 (2003)
10. CMMI Product Team: CMMI for Software Engineering, Version 1.1, Continuous Representation (CMMI-SW, V1.1, Continuous). Software Engineering Institute, Pittsburgh, PA (2002)
11. CMMI Product Team: CMMI for Software Engineering, Version 1.1, Staged Representation (CMMI-SW, V1.1, Staged) Software Engineering Institute, Pittsburgh, PA (2002)
12. Conboy, K., Morgan, L.: Beyond the customer: Opening the agile systems development process. Information and Software Technology 53 (2011)
13. Conboy, K., Wang, X., Fitzgerald, B.: Creativity in Agile Systems Development: A Literature Review. In: Dhillon, G., Stahl, B.C., Baskerville, R. (eds.) CreativeSME 2009. IFIP AICT, vol. 301, pp. 122–134. Springer, Heidelberg (2009)
14. Couger, J.D.: Creative problem solving and opportunity finding. Decision making in operations management series. Boyd & Fraser Pub. Co., Hinsdale (1994)
15. Couger, J.D., Dengate, G.: Measurement of Creativity of I.S. Products. In: Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, vol. 4, pp. 288–298 (1992)
16. Couger, J.D., Higgins, L.F., McIntyre, S.C.: (Un)Structured Creativity in Information Systems Organizations. MIS Quarterly 17, 375–397 (1993)
17. Dewey, J.: Human nature and conduct: An introduction to social psychology. The Modern Library, New York (1957)
18. Gibson, J.J.: The Theory of Affordances. In: Shaw, R., Bransford, J. (eds.) Perceiving, Acting, and Knowing - Toward an Ecological Psychology, pp. 67–82. Lawrence Erlbaum Associates, Publishers, Hillsdale (1977)
19. Gorschek, T., Fricker, S., Palm, K., Kunsman, S.A.: A Lightweight Innovation Process for Software-Intensive Product Development. IEEE Software 27, 37–45 (2010)
20. Hey, J.H.G.: Framing innovation: negotiating shared frames during early design phases. Journal of Design Research 6, 79–99 (2007)
21. Highsmith, J., Cockburn, A.: Agile software development: the business of innovation. Computer 34, 120–127 (2001)
22. Hirschheim, R.A., Klein, H.K., Lyytinen, K.: Information systems development and data modeling: conceptual and philosophical foundations. Cambridge University Press, Cambridge (1995)
23. Larman, C.: Agile and iterative development: a manager's guide. Addison-Wesley, Boston (2004)

24. Mahaux, M., Maiden, N.: Theater Improvisers Know the Requirements Game. IEEE Software 25, 68–69 (2008)
25. Maiden, N., Ncube, C., Robertson, S.: Can Requirements Be Creative? Experiences with an Enhanced Air Space Management System. In: 29th International Conference on Software Engineering, ICSE 2007, pp. 632–641 (2007)
26. Maiden, N., Robertson, S.: Integrating Creativity into Requirements Processes: Experiences with an Air Traffic Management System. In: Proceedings of the 2005 13th IEEE International Conference on Requirements Engineering (RE 2005), pp. 105–114 (2005)
27. McCall, R.: Critical Conversations: Feedback as a Stimulus to Creativity in Software Design. Human Technology 6, 11–37 (2010)
28. Misra, S.C., Kumar, V., Kumar, U., Mishra, R.: Goal-Driven Measurement Framework for Software Innovation Process. Journal of Information Technology Management XVI, 30–42 (2005)
29. Naur, P., Randell, B. (eds.): Software Engineering: Report on a Conference sponsored by the NATO Science Committee, Garmisch, October 7-11, 1968. Scientific Affairs Division, NATO, Brussels (1969)
30. Pikkarainen, M., Codenie, W., Boucart, N., Alvaro, J.A.H. (eds.): The Art of Software Innovation - Eight Practice Areas to Inspire your Business. Springer, Heidelberg (2011)
31. Pressman, R.S.: Software engineering: a practitioner's approach. McGraw-Hill Higher Education, Boston (2005)
32. Rajlich, V.: Changing the paradigm of software engineering. Commun. ACM 49, 67–70 (2006)
33. Rose, J.: Software Innovation: Eight work-style heuristics for creative system developers. Software Innovation, Aalborg University, Department of Computer Science (2010)
34. Schön, D.: Designing as reflective conversation with the materials of a design situation. Knowledge-Based Systems 5, 3–14 (1992)
35. Schön, D.A., Wiggins, G.: Kinds of seeing and their functions in designing. Design Studies 13, 135–156 (1992)
36. Shneiderman, B.: Creativity support tools: accelerating discovery and innovation. Communications of the ACM 50, 20–32 (2007)
37. Sternberg, R.J., Lubart, T.I.: The concept of creativity: Prospects and paradigms. In: Handbook of Creativity, pp. 3–15. Cambridge University Press, Cambridge (1999)
38. Sutcliffe, A.: Juxtaposing Design Representations for Creativity. Human Technology 6, 38–54 (2010)
39. Swanson, E.B., Ramiller, N.C.: Innovating Mindfully with Information Technology. MIS Quarterly 28, 553–583 (2004)
40. Tidd, J., Bessant, J.R., Pavitt, K.: Managing innovation: integrating technological, market and organization change. Wiley, Hoboken (2005)
41. Turner, P.: Affordance as context. Interacting with Computers 17, 787–800 (2005)
42. Wang, J., Farooq, U., Carroll, J.M.: Does Design Rationale Enhance Creativity? Human Technology 6, 129–149 (2010)
43. Wenger, E.: Communities of practice: Learning, meaning, and identity. Cambridge University Press, Cambridge (1998)
44. West, D., Solano, M.: Metaphors be with you! (Metaphor System). In: Proceedings of the Agile Development Conference, pp. 3–11 (2005)

The Facets of Sociomateriality: A Systematic Mapping of Emerging Concepts and Definitions

Elena Parmiggiani and Marius Mikalsen

Norwegian University of Science and Technology
Department of Computer and Information Science, Sem Sælands vei 7-9,
NO-7491, Trondheim, Norway
{parmiggi,mariusmi}@idi.ntnu.no

Abstract. Sociomateriality is gaining momentum and is by now characterized as a research stream in the information system field. Although some definitions emerged, there is still uncertainty about how to conceptually and analytically address sociomateriality. The debate ranges from understanding sociomateriality as just a fancy word for technology to treating it as a de-facto theory of the human-technology relationship. To bring the field forward, a common basic understanding of what sociomateriality entails is needed. In this paper we set out to contribute to such an understanding. We do this by conducting a systematic mapping study of emerging concepts and definitions in the current empirical body of literature on sociomateriality. Our analysis finds three key resulting facets: mutuality (*what* is a sociomaterial assemblage?), performativity (*how* does it perform?), and multidimensionality (*When* and *where* does it perform?). Our findings outline how sociomaterial studies analytically and methodologically address performativity spanning across time and space.

Keywords: Sociomateriality, Systematic Mapping, Mutuality, Performativity, Multidimensionality.

1 Introduction

Since Orlikowski and Scott's call for studies that address the "constitutive entanglement" of the material and the social [1], the concept of *sociomateriality* has gained momentum in the information systems (IS) literature, and constitutes by now a significant "wave of research" [2]. The term has, in the other variants "socio material" or "socio-material", been around for a long time. In 1979, Østerberg and Vale [3:75] explain "sociomatter" as consisting of "human beings and things in a useful context". While Østerberg and Vale's analysis is on the level of society and the crucial distinction between those who own ("matter" and consequently lives in a "sociomaterial society") and those who do not own, another even earlier stream of research focused on the role of technology in organizational change. Socio-technical theorists argued already in the 1950s that technological change implicated both the material and the social [4].

By building on these early insights on the relation between technology and organization, IS research has long recognized the importance of both the material and

M. Aanestad and T. Bratteteig (Eds.): SCIS 2013, LNBIP 156, pp. 87–103, 2013.

the social aspects of the human-technology relationship. Ample empirical evidence shows the way new technologies alter the "social dynamics" of organizations [5], be that change in organizational structures, decision-making, and power relationships in formal organizations [3], or change in informal communication networks [6]. Seminal work by Orlikowski [7] indicates technology as a crucial amplifier for restructuring organizations. The other way around, research also documents that technology is not "written in stone", and rather shows the malleability of technology, explaining how technologies emerge as products of a social process; negotiations, human agency, and personal interest [8]. Information systems, when put in use, are also subject to a great deal of local workarounds, improvisations, and tinkering [9].

The bi-directional relationship between the material and the social is properly established. But we have as of yet not revealed all of its subtle nature; "what is lacking is a satisfactory account of the interwoven relationship between IT and organizational transformation (…) we need to learn more about *how* this interplay works, not only that it exists" [10:326] and resolve "the epistemological and ontological nature of the relationship between the material and the social" [8:160]. The research wave on sociomateriality aims to do just that.

In this paper we report the first findings from our systematic mapping study of the growing body of literature on sociomateriality. Our motivation to do such a study was that in order to release the potential explanatory power of sociomateriality we need to have a base definition and understanding of the term. To do so, we explored how sociomateriality is used in empirical studies. Currently, the debates range from some characterizing sociomateriality as simply (yet another) fancy academic word for technology, to others treating it as a de facto theory of human-technology relationships. We have no wish or intention on concluding this debate. Rather through a systematic mapping study of the body of literature, we aim to add to Leonardi's initial definition of sociomateriality [11] (see section 2). We do so by inductively deriving three possible themes (or *facets*) characterizing empirical sociomaterial research that should be part of the future sociomaterial discourse and form parts of a base definition that can bring more understanding to the field.

This work does not aim at taking sides in a debate on sociomateriality. We are not arguing in favor or against the need to take a sociomaterial approach rather than building on different research agendas. Rather, we register that a growing body of literature in IS subscribing to this approach, and therefore we make an effort to depict its characteristics and implications. We are aware that other strands of research are also looking at the same challenges (e.g. materialist theories and technology studies in feminist technoscience – see e.g. [12]) with different terminologies. However, for the purposes of this review we chose to focus on the literature that explicitly addresses sociomateriality within the IS field.

The rest of this article is organized as follows. In section 2 we explore the origins of sociomateriality, its status, and explain the rationale for studying it. In section 3 we introduce the systematic mapping method we use in our study, and how we through a 5-step procedure with defined exclusion/inclusion criteria went from a total of N=937 studies to N=51 studies subject for analysis. In section 4 we analyze the studies and map out three facets that are emerging in the literature: *mutuality*, *performativity*, *multidimensionality*. Finally we conclude with some considerations on future research directions.

2 The Sociomaterial Rationale

Arguably being at the center of IS attention, explaining the relationship between the social and material is an intriguing challenge, but it has proven to be a difficult one. The same way that socio-technical theorists soon begun to focus purely on social interventions [4], working on explaining the nature of the sociomaterial relationship has led researchers to tilt towards either focusing on the social (organization and process) or the material (technology and other objects). Leonardi and Barley [8:160] suggest that the reason for us tilting in either direction is because we "conflate two important but separate, philosophical distinctions: the difference between determinism and voluntarism, on one hand, and the distinction between materialism and idealism, on the other." The challenge at hand then is to acknowledge that materiality matters, while still assuming that humans perform agency and execute free will, and that this can and is used to also shape and form the material.

Sociomateriality has gained popularity by challenging the separation between technology, work, and organization altogether. Contractor et al. [5] sees sociomateriality as an "analytical break" that can help us avoid the dichotomy that exists between the social and the technical. A sociomaterial understanding "… asserts that materiality is integral to organizing, positing that the social and the material are constitutively entangled in everyday life. A position of constitutive entanglement does not privilege either humans or technology (in one-way interactions), nor does it link them through a form of mutual reciprocation (in two-way interactions). Instead, the social and the material are considered to be inextricably related – there is no social that is not also material, and no material that is not also social." [ibid., p. 41].

Many scholars have contributed to the understanding leading up to the notion of *sociomaterial constitutive entanglement*. Crinson [13] explains how Orlikowski [14] in formulating the sociomaterial agenda builds on Latour's actor-network theory (ANT) [15], Knorr-Cetina's concept of object-centered sociality [16], Bijker's concept of sociotechnical ensemble [17], Law's concept of relational materiality [18], and Beunza et al.'s concept of material sociology [19]. The ANT affiliation of sociomateriality is also established by Björgvinsson et al. [20:102] explaining how "these kinds of socio-material assemblies that Bruno Latour so strikingly has characterized as collectives of humans and nonhumans". However, ANT is grounded on the assumption that humans and nonhumans pre-exist the establishment of collectives. There is a fundamental ontological distinction with e.g. Karen Barad's notion of agential realism [12], where humans and nonhumans do not pre-exist, but are rather constituted as the entanglements are configured. Matter therefore becomes an active agent in that, by materializing, it performs an action.

Barad [12] explains how the observation of the constitutive entanglement between phenomena and material arrangements influence research on sociomateriality, but she also notes that most of sociomaterial empirical work has focused on the constitutive entanglement of computers and work, as postulated by e.g. Suchman [21].

As intriguing as it may be to study, analyze, and theorize within the sociomaterial research stream, it is a tall order no doubt, to not only bridge determinism and agency, and the material and social, but to build a new understanding where social and

material are "*constitutively entangled",* becoming sociomaterial ensembles (paraphrasing Bijker [17]). The challenging nature of the task has led to ontological confusion [11] where the terms sociomaterial and socio-technical for instance have been used interchangeably, and the term sociomaterial has been used to simply signify that there is a bi-directional relationship without properly exploring it. Taking a first step towards finding a clarification, Leonardi [ibid.] suggests a "rough and tentative" glossary of terms that aims to "…begin a movement in the direction of clarity so that scholars use the terms productively to theorize the complexity of collective endeavors, generally, and organizational dynamics specifically". In particular, Leonardi differentiates a socio-technical system from a sociomaterial one. The former is defined as the "recursive (not simultaneous) shaping of abstract social constructs and a technical infrastructure that includes technology's materiality and people's localized responses to it". The latter is instead characterized in terms of the "*enactment* of a particular set of activities that meld materiality with institutions, norms, discourses, and all other phenomena we typically define as "social.""

Taking such first steps towards ontological clarity is certainly a prerequisite to theorize. But as Constantinides and Barret [2:291] explain: "there is still the problem of how to study the constitutive entanglement of the social and the material; where does one start, methodologically and analytically, to trace the entanglement". The diverse origins of sociomateriality, the first attempts at definitions of the term, and the greenfield area of methodology and analysis taken together is what motivated us to do a systematic mapping study of this emerging research field.

3 Method

3.1 Systematic Mapping

To the best of our knowledge (we have searched and talked to experts), no systematic mapping study has been undertaken on the topic of sociomateriality. The systematic mapping method we have applied is a combination of a literature review as known in the IS field [22, 23] and a systematic mapping process. The latter is a protocol known from the domain of software engineering [24]. The stages of the methodology are the following: first we defined the research question. Based on that, we set a search string and query a selection of major databases available online. We then selected articles through a set of defined steps by applying predefined exclusion criteria. For the sake of traceability, we created separate EndNote database at each step. See Fig. 1 for an outline of the process steps and outcomes (figure adapted from Petersen et al. [24]).

3.2 Definition of Research Questions

Our main goal is that of gaining a clearer overview of the evidence around sociomateriality. We therefore seek to answer the following broad research question: ***What is empirically known about sociomateriality?***

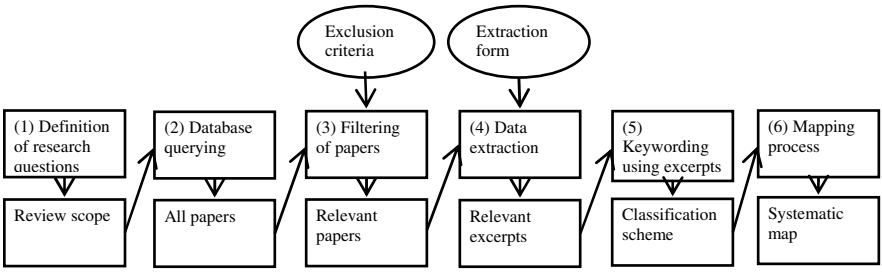

Fig. 1. The systematic mapping protocol applied

3.3 Database Querying

Based on our research question, the following search string was defined: ***sociomaterial* OR "socio-material*" OR "socio material*"***, to fetch all the literature addressing sociomateriality.

We adopted an open search to find all articles addressing the topic. Titles, abstracts, keywords, and full texts were searched. We chose a range of well-known scholarly databases from the list provided by Levy and Ellis [23], in order to include contributions from all relevant IS journals and conference proceedings (see Table 1). We selected all the conference and journal papers published until 01.01 2013.

Table 1. Online source and number of articles retrieved

Source	*# of results*
ISI Web of Knowledge	114
AIS Electronic Library (AISeL)	121
JSTOR	193
ProQuest - ABI/INFORM	283
ScienceDirect	179
SpringerLink	152
TOTAL	**1042**
TOTAL after duplicate removal	**937**

3.4 Filtering of Papers and Data Extraction

From the results of step 2, we manually analyzed the article set through a sequence of pre-defined steps (see Fig. 2). In step 1, we created an EndNote database schema to collect the titles, authors, reference type, abstracts, and (if available) keywords resulting from the above search. We gathered a total of 937 entries after we automatically removed duplicate records. At step 2, 3, and 4, we selected papers eligible for inclusion in a process of increasing levels of granularity. We first considered the titles, then the abstracts and finally the full texts. If abstracts were not available, we read the full text of papers.

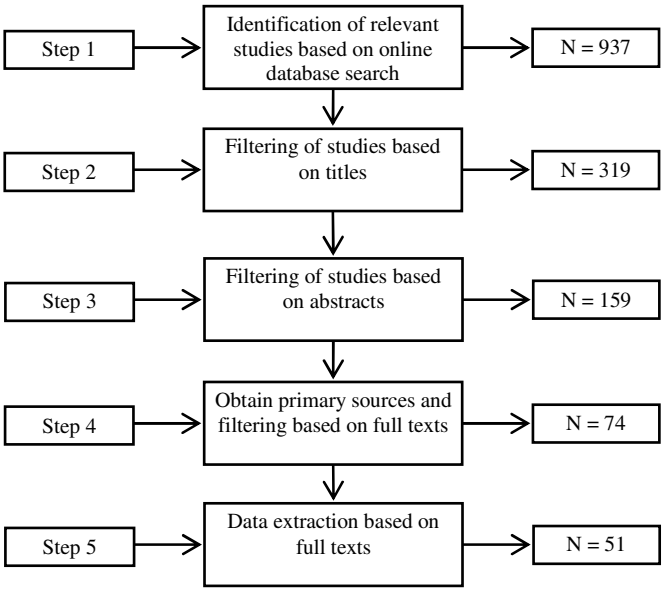

Fig. 2. Steps of the article filtering process (zooming into steps (3) and (4) of **Fig. 1**)

At each stage, studies were excluded by meeting one of the two following criteria:

- [*Relevance*] Contributions must clearly belong to the research areas of information systems, software engineering, or computer science;
- [*Rigour*] Contributions should provide empirical findings (e.g. by describing case studies) of the concept of sociomateriality. For instance, those studies that explicitly declared to be literature reviews in the abstract or that were lacking a proper paragraph describing the empirical methodology adopted were discarded.

At step 5 we extracted data from 74 studies according to a pre-defined extraction form adapted from systematic reviews in software engineering [25]. This step enabled us to further filter the corpus of relevant articles. While extracting, we applied an additional criterion based on the *relevance* of the studies for the research stream on sociomateriality. We excluded those articles that either:

- Did not provide an explicit definition of sociomateriality;
- OR did not refer to relevant sociomateriality literature in information systems;
- OR did not relate to primary results/findings.

This was done to remove articles that used the term sociomateriality without defining it or pointing to earlier sociomaterial research, indicating that they use the term assuming it to be well understood and agreed upon, or simply as an indicator that the social and material is entangled (which is known). The outcome of this stage was a final corpus of 51 primary studies.

3.5 Keywording and Mapping Process

Based on data extraction forms, as Webster and Watson [22] recommend, we adopted a concept-centric approach that is likely to better synthetize the literature and thus achieving a mapping framework. We identified a set of concepts by means of a keywording strategy (as discussed by Petersen et al. [24]) applying an open coding technique [26] to the abstracts of the relevant papers found in the previous step. We then clustered the keywords in order to define categories. In line with an interpretivist research tradition [27], our gradual understanding emerged through an iterative creation of categories. To increase validity, the categories were discussed between the two authors and with other members of the research group. The interpretive approach led to a classification scheme – a *systematic map* – made of three facets investigating the *what, how, and where/when of sociomateriality.*

1) **Facet 1 – *What***: What definition of sociomateriality is provided? Which theoretical or conceptual backgrounds are leveraged to inform it?
2) **Facet 2 – *How***: How are the sociomaterial entanglements understood to perform in practice? What does it mean concretely for researchers to follow up on a case through a sociomaterial lens?
3) **Facet 3 – *Where/When***: When and where do sociomaterial assemblages stretch and shape across time and space?

Using these top levels as a map, the relevant articles were analyzed and interpreted into our classification scheme.

4 Results

4.1 Sociomateriality as an Ex-Post Label

As Monteiro et al. assert [28:91] "rather than an independent set of concepts, sociomateriality summarizes and highlights salient aspects and insights gained in information systems research over the last couple of decades". However, "sociomateriality states *that* use/technology is entangled" (ibid., p. 92, emphasis in original), but not *how*. Along the same lines, Contractor et al. [5:685] state that the sociomaterial approach "does not provide much guidance in specifying how researchers might depict sociomaterial relations empirically" . What is to be searched for is a thorough vocabulary to describe *what* constitutes the entanglements (facet 1); *how* they play out in practice – or perform (facet 2); and *under which circumstances* (facet 3). We therefore divided the primary studies into two metacategories (see Table 2). This revealed that almost 51% of the studies (metacategory 1) are actually using the term "sociomaterial" as an ex-post label. They could be filed under the "nominal" category used by Orlikowski and Iacono [29] to collect those studies invoking the relevance of the technological element, but not theorizing or conceptualizing it. 49% (metacategory 2) examine the concept of sociomateriality by going further than citing Wanda Orlikowski and Lucy Suchman's milestone pieces of work [1, 14, 21, 30]. As a result of the keywording and mapping process described in section 3.5, we

inductively derived three conceptual categories (facets) and finally grouped the contributions that belong to metacategory 2 accordingly (see Table 3).

Table 2. Distribution of studies on the two metacategories

Metacategory	*Description*	*# of articles*
1	Illustrating an empirical phenomenon under the label of sociomateriality.	26/51 (51%)
2	Providing a theoretical or methodological contribution to the understanding of sociomateriality.	25/51 (49%)

4.2 Facet 1: Mutuality – The WHAT of Sociomateriality

Mutuality. Baptista et al. [31:172] write: "More recently the IS literature has suggested that as technology becomes more intricate to the functioning of organizations and to the routine behaviors of employees, the social and technical dimensions develop to mutually constitute the 'sociomateriality' of an organization [14]. This new conceptualization in IS research has raised subtle but relevant questions, for example about the ontological separation between technology and the social context that influences its use". Kuk and Davies [32] put a stress on the element of mutuality and define the sociomaterial lens as one that "draws attention to the mutually constituted nature of both human and material agency [33:4], and the roles that social and material artifacts play". Barley et al. [4] add on to this definition by considering the properties of technology as entangled also with social norms, individual interpretations, and work flows. Riemer and Johnston [34] describe mutual influences in sociomaterial entanglements as a circularity of reference.

Symmetry. Five articles are explicitly grounding a definition of sociomateriality in its actor-network theory (ANT) roots [35, 36, 37, 38, 39], for instance by extensively referring to the work of Bruno Latour or John Law [15, 40]. ANT represents a powerful – perhaps overly exploited – vocabulary at analyzing sociomaterial constitutive entanglements [2, 36, 39, 41, 42]. Al-Mahmood [39] focuses on the Latourian bases of ANT to introduce sociomaterial assemblages as networks of people, nature, and things. In order to underline the relational perspective of a technology, Almklov et al. [9] draw the trajectory of sociomateriality from research in science and technology studies, social informatics (highly inspired by ANT and anchored to Monteiro and Hanseth [10]) towards Orlikowski and Scott [1]'s developments over structuration theory.

Imbrication of Agencies. Other studies take Leonardi's work as a starting point by drawing on his notion of "imbrication" [11], for instance [42, 43, 44]. This concept is more geared towards the role of the interweaving agencies that produce a result (the term is borrowed from the names of the interlocked roof tiles used in ancient Roman and Greek roofs). Imbrications are illustrated as networks of human and nonhuman elements. This aspect is also at the core of the definition of *information infrastructures* provided by Star and Ruhleder [45], who present infrastructures as a

relational concept that emerges in situ through organized practices, a feature explicitly indicated in two studies we retrieved [9, 43]. Adding on the definition of imbrications, Bratteteig and Verne [43] make an interesting attempt at bridging the concepts of "sociomateriality" and "imbrication". The authors add to the view of sociomaterial entanglement of heterogeneous aspects the concept of "imbrication", to analytically "disentangle" the knots within sociomaterial assemblages at different levels of complexity. Zorina and Avison [44] instead address the influence of inter-organizational imbrications in Web 2.0 communities. Introna and Hayes [42:120] reshape notions of "formative context" from Ciborra [46] and "technological frames" from Contractor et al. [5] under the interpretive research tradition and an ANT perspective, to underline how technology and humans constantly frame each other within a sociomaterial nexus: "In our imbrications with technology we are their constitutive contexts as much as they are our constitutive context".

4.3 Facet 2: Performativity – The HOW of Sociomaterial Assemblages

Analytical Disentanglements. According to Orlikowski and Scott [33] it is possible to untie the knots of sociomaterial assemblages only analytically. Bratteteig and Verne [43] contribute by suggesting a matrix of concepts to solve a sociomaterial entanglement in complex daily situation. But how does the analytical "disentanglement" of sociomaterial assemblages happen in literature? From the previous facet we found a clear emergence of a strong ANT root in how sociomateriality is defined in empirical case studies. Some studies make a step forward by adopting an outlook based on the later versions of ANT. The work by Thompson [37] is relevant in that it uses ANT to explore online work-learning practices. Influenced by Mol [47], the author describes the different ways through which learning practices are enacted thanks to heterogeneous "socialities" and "materialities" that however lead to the same final outcome. Introna and Hayes [42] subscribe to the interpretivist tradition that sees a co-constitutive relation between the context as a whole and the parts as the "texts". According to these authors, it is however fundamental to underline how technical elements are not necessarily "texts" to be studied with a "context" made of human values and assumptions. This complies with Latour's claim that there is no clear-cut distinction between subjects and objects. A subset of the primary studies that we took into consideration contribute by figuring out the role of elements that are neither strictly human not technical, but have been often taken for granted or left lingering in the background in previous literature. For instance, a few studies underline the role of norms [48], human motivations in action [32], historical and cultural traditions [42], local/global contingencies [35, 48, 49].

The Performativity of Sociomateriality. So far, we have tried to delineate how literature addresses the elements of a sociomaterial assemblage. However, one of the tenets of sociomateriality is performativity, explained by Barad [12] as the enactment of a specific configuration of a reality. A few articles explicitly address the *performative* aspects of sociomaterial entanglements [2, 9, 28, 32, 50]. Monteiro et al. [28] dismiss a representational view of the entangled technological elements in favor

of a performative one: the relevance of this piece lies in its provision of a set of labels to depict the practices through which the operators of an oil and gas company cope daily with the entangled representations of the ICT elements. The same perspective is also adopted by Almklov et al. [9], that show how technologically-enabled representations are actually the result of empirically driven representational practices rather than passive readings of sensors. The information modeling activities, together with the role of experience, are finely illustrated as pragmatic sociomaterial construct that emerge in practice. Another interesting contribution is given by Kuk and Davies [32] in a study following the process involved in the liberation and use of Linked Open Data. Open Data are seen as having their intrinsic value that emerges through the practical interlocking of both human and artifact's agency. The authors argue how not only the performativity of artifacts (e.g., Open Data themselves), but also human motivation can be leveraged to drive innovation in public services. A novel approach to address sociomaterial entanglements in practice is proposed by Constantinides and Barrett [2], who adopt a methodology based on narrative networks to investigate the multiple possibilities of enacted coordinated practices.

Empirical Methodology: Data Collection. A sociomaterial perspective can be innovative at least at two different levels: as a theoretical concept and as a research method. The choice of a specific research design affects the way the researcher is able to account for the performativity of an object of study as a sociomaterial assemblage in its unfolding. One of the emerging characteristic of the empirical studies conducted with a sociomaterial approach is the way data collection is carried out. All the contributions we found were designed as qualitative case studies, mainly based on ethnographies. As such, they mostly rely on interviews, documentation, and observations. In addition, we registered an emerging variation in the typologies of data retrieved. For instance, Kuk and Davies [32] consider Twitter messages in their analysis of the development of Linked Open Data. Al-Mahmood [39] and Van Osch and Mendelson [51] take instead multimedia into account.

4.4 Facet 3: Multidimensionality – The WHERE and WHEN of Performativity

A number of studies among those that we gathered also contribute to the notion of sociomateriality by expanding the concept along the dimensions of time and space. The motivation for this shift follows directly from the analysis of the practical performances of sociomaterial assemblages.

Local vs. Global. In most of the cases found in the primary studies, the observed empirical phenomenon is a longitudinal stretching of work practices. Observations often unveil that modern work flows are spanning across several geographical locations but must follow either too generic or context-specific norms and process models [42, 49], and are ultimately performed through situated, ad-hoc local practices [2, 35]. The result of these observations is that a continuous "bouncing" effect is created between local and non-local (or global) concerns and the sociomaterial entanglements are therefore augmented not only along the space dimension, but also that of time. Among the studies trying to provide an explanatory framework to this

further complexity is that by Nicolini [38]. Here, the author presents a lexicon (based on zooming-in and zooming-out practices) for "recursively navigating between local instances and their connections" (p. 1412). Zorina and Avison [44] argue for the need to address the external context of inter-organizational imbrications to understand contemporary organizations. In his application of the narrative-network approach to the analysis of sociomaterial assemblages, Constantinides and Barrett [2] demonstrate how different interconnected practices can be traced in time and space. Johri [35] uses the concept of "*sociomaterial bricolage*" by gathering Orlikowski and Scott [1]'s definition and that of *bricolage* [46] as a perspective to understand situated routines during location-spanning work practices. Other studies set instead the magnifying lens on the unfitting of social and technical assumptions inscribed with a model or a technology, that thus struggle to adapt to either local emerging realities or to more global factors. Introna and Hayes [42] tell how the development of a software to detect plagiarism within British educational institution misinterprets the learning habits of Greek students (geographical dimension) that are due to their studying practices traditionally developed (time dimension) in their home country. Monteiro and Rolland [49] specifically tackle the space dimension by introducing the concept of commensurability, to trace the similarity between trans-situated sociomaterial practices. Along the same lines, Monteiro et al. [50] analyze how similarity between technologically mediated work practices is achieved as a political, pragmatic, and performative process.

Settings of Case Studies. The shift to the performative nature of practices entailed by sociomateriality leads to the re-definition of the traditional concept of "space" as commonly conceived in longitudinal case studies. Typical case studies in the information system literature have been carried on inside organizations – or hospitals – within working settings. Even though the majority of the studies we gathered are still oriented towards the workplace, a relevant discovery of our review is the widening spectrum of heterogeneous cases that scholars have recently started to follow. A sociomaterial perspective seems therefore to enable the enlargement of the research scope to scenarios that were previously ignored. For instance, Kuk and Davies [32] conduct an ethnography of the data liberation process by the British government, and of the way hackers got hold of the data to trigger a process to turn them into Linked Open Data. The Web 2.0 is also emerging as a natural scenario for adopting a sociomaterial lens. Scott and Orlikowski [56] aim to understand what they call the "sociomateriality of accountability" in the context of social media, specifically the TripAdvisor website. The learning experience of workers though online communities is investigated by Thompson [37], whereas Nicolini [38] addresses the field of telemedicine. Sociomaterial interactions embedded in multimedia technologies are also addressed by Van Osch and Mendelson [51] within a community and a primary school.

Table 3. Overview of the studies in metacategory 2 and the facets they present. Relevant features in brakets.

Article	*Facet 1 (What)*	*Facet 2 (How)*	*Facet 3 (When/Where)*
Constantinides and Barrett [2]	x	x (performativity)	x (narrative networks to trace in time and space)
Contractor et al. [5]	x	x (multidimensional networks)	
Almklov et al. [9]	x	x (performativity of representations and models)	
Orlikowski [14]	x	x	
Monteiro et al. [28]	x	x (performativity)	
Kuk and Davies [32]	x	x (performativity; Tweets as data)	x (Case study: Linked Open Data)
Johri [35]	x (ANT and other)	x	x (sociomaterial bricolage; local vs. global)
Gasson [36]	x (ANT)	x (political attachment and local group mobilization in misalignments)	
Thompson [37]	x (later ANT)	x (performativity of learning practices)	x (Case study: online learning)
Nicolini [38]	x (ANT and other)	x	x (zoom-in/zoom-out)
Al-Mahmood [39]	x (ANT)	x (multimedia as data)	x (online learning; spaces emerging relationally)
Awazu and Newell [41]	x	x (knowing through practice)	x (implementation as cultural and historical practice)
Introna and Hayes [42]	x	x (historical and cultural traditions)	x (global practices vs. local contexts)
Bratteteig and Verne [43]	x	x (entanglements + imbrications)	
Zorina and Avison [44]	x		x (inter-organizational imbrications)
Leclerq et al. [48]	x	x (norms)	
Monteiro and Rolland [49]	x	x (performativity)	x (trans-situadedness)
Monteiro et al. [50]	x	x (performativity)	x (family resemblance of distributed practices)
Van Osch and Mendelson [51]	x	x (multimedia as data)	x (Case study: multimedia)
Orlikowski [52]	x (materiality as scaffolding)		
Østerlie et al. [53]	x	x (dual materiality; knowing emergent from different levels of materiality)	
Baptista et al. [31]	x	x (institutionalization of technology)	
O'Farrell et al. [54]	x		
Riemer and Johnston [34]	x (circular reference)		
Svahn et al. [55]	x		

5 Discussion and Concluding Remarks

There is a long tradition among sociotechnical scholars to acknowledge the role of technologies within society and in particular organizations. *Sociomateriality* emerged as an agenda to gather the streams of research aiming to account for the emergent interplay between social and material aspects. We queried six major online scholarly databases to retrieve articles providing and discussing empirical findings related to the concept of sociomateriality. After a filtering process, we were left with 51 primary

studies. 25 of them were considered to go beyond acknowledging the entanglement of the social and the material and to flesh out the conceptual and analytical details of sociomateriality. Our analysis yielded three main facets to outline the emerging core characteristics that extend the tentative glossary of terms by Leonardi [11]. From a first analysis of the attributes of the three facets, we derive the following connections with adjacent IS research streams.

First of all, the key feature of sociomaterial assemblages is their *performativity*. Subscribing to a sociomaterial agenda means addressing them as emerging in practice within a specific context. Actor-network theory (ANT) has provided a powerful vocabulary to do this, by ad-hoc opening or closing the black boxes inside an actor network. Researchers applying this perspective draw on a long tradition of acknowledging the importance of describing assemblages (or actor networks) through their performative stance. Such appraisal is keen on recognizing the malleability of materiality and the workarounds enacted by humans in situated settings and has therefore a consequence on the way the elements of an assemblage are treated analytically (see facet 2). However, studies that are taking this view tend to drift along the peculiar symmetry that the first ANT imposes on the elements of an actor network, by bringing social and the material elements to the same level (see facet 1). The tendency to foreground the role of the technological element is, at least partly, in line with the arguments raised almost two decades ago by Monteiro and Hanseth [10], and a few years later by Orlikowski and Iacono [29] and Orlikowski [52] asking for a deeper comprehension of the specific role of technology and more in general of materiality in order to understand how knowledge flows within organizations. The few studies that are able to adopt the insight of the later versions of ANT [47] better account for the mutable interplay between more or less visible actors and their relationships distributed in time and space.

Second, sociomateriality has also consequences on the *design of empirical research*, in terms of the settings where studies are conducted (facet 3) and the data collection process (facet 2). ANT-based perspectives still play a major role, primarily due to ANT being a more mature theoretical scaffold that has been evolving during the last 30 years. Researchers adopting an ANT outlook are thus more accustomed to its tenets. Moving forward, we depicted the recent shift from workplace-oriented case studies towards more heterogeneous settings, embracing e.g. multimedia and social networks (facet 3). It is a recent trend registered in literature also outside the scope of our review, see for instance Nardi's ethnographies of online gaming practices [57], Knorr-Cetina's study on the encounter between energy physics and molecular biology [16], and Barad's account for social and natural meanings in the universe [13]. The acknowledgement of the performativity of an assemblage broadens the notion of "space" from the physical location to encompass virtual or distributed settings. In addition, what is considered as *data* has expanded, to include e.g. messages on social networks and multimedia. Indeed, the adoption of a sociomaterial lens implies the need to reconceive the understanding of "objects" (also when intended as technologies). This is consistent with Barad's claim that materiality does not always equal computers, as the IS field has traditionally related to workplace settings [13]. The notion is compatible with that of the Internet of Things, intended as networked

heterogeneous devices and tools. This wider view underscores the role of context: each artifacts is applied inside different material and cultural practices [29].

Third, the definition of sociomateriality provided by Leonardi [11] is very close to that of information infrastructure [10]. The latter is indeed characterized by features that go beyond the technology and stretch at different practical, institutional, and organizational scales with different temporal concerns [58]. A sociomaterial account might offer a tool to embrace these elements if it successfully applies all the facets outlined above. This is in line with the recent literature in IS. For instance, Barley et al. [4] call for an improved temporal and spatial understanding of workplace technologies, often too bond to the implementation and design dynamics. Karasti et al. [59] also demonstrate how the long-term matters are equally important to the short-term one. By exploring the temporal dimension (facet 3), sociomateriality focuses on how knowing in an information infrastructure is not only a situated performance, but also emerges as a performative accomplishment. It arises from the interplay between particular configurations of not only material phenomena, but also the material arrangements set up by humans to discover these phenomena and the knowledge practices established in time [53].

We have in this work taken another step towards a definition of the sociomaterial agenda. Our mapping of emergent concepts suggests that future studies should focus on aspects of mutuality, performativity, and multidimensionality. Then they will provide a base to account for how associations of humans and nonhumans are dynamically articulated [60]. In so doing, sociomaterial studies can be relevant also towards notions from agential realism and feminist technoscience.

References

1. Orlikowski, W.J., Scott, S.V.: Sociomateriality: Challenging the Separation of Technology, Work and Organization. The Academy of Management Annals 2(1), 433–474 (2008)
2. Constantinides, P., Barrett, M.: A narrative networks approach to understanding coordination practices in emergency response. Information and Organization 22(4), 273–294 (2012)
3. Østerberg, D., Vale, M.: Historical Materialism and Durkheim's Collective Logic. International Journal of Sociology 9(4), 75–89 (1979)
4. Barley, S.R., Meyerson, D.E., Grodal, S.: E-mail as a Source and Symbol of Stress. Organization Science 22(4), 887–906 (2011)
5. Contractor, N.S., Monge, P.R., Leonardi, P.M.: Multidimensional Networks and the Dynamics of Sociomateriality: Bringing Technology Inside the Network. International Journal of Communication 5, 682–720 (2011)
6. Latour, B.: Literature, in Science in Action: How to follow Scientists and Engineers through Society. Harvard University Press, Cambridge (1987)
7. Orlikowski, W.J.: Integrated information environment or matrix of control? The contradictory implications of information technology. Accounting, Management and Information Technologies 1(1), 9–42 (1991)
8. Leonardi, P.M., Barley, S.R.: Materiality and change: Challenges to building better theory about technology and organizing. Inf. Organ. 18(3), 159–176 (2008)

9. Almklov, P.G., Osterlie, T., Haavik, T.K.: Punctuation and extrapolation: representing a subsurface oil reservoir. Journal of Experimental & Theoretical Artificial Intelligence 24(3), 329–350 (2012)
10. Monteiro, E., Hanseth, O.: Social shaping of information infrastructure: on being specific about the technology. In: Orlikowski, W.J., et al. (eds.) Information Technology and Changes in Organizational Work, pp. 325–343. Chapman & Hall, London (1996)
11. Leonardi, P.M.: Materiality, Sociomateriality, and Socio-technical Systems: What Do These Terms Mean? How Are They Different? Do We Need Them? In: Leonardi, P.M., Nardi, B.A., Kallinikos, J. (eds.) Materiality and Organizing: Social Interaction in a Technological World. Oxford University Press, Oxford (2012)
12. Barad, K.: Posthumanist Performativity: Towards an Understanding of How Matter Comes to Matter. Journal of Women in Culture and Society 28(3), 801–831 (2003)
13. Crinson, I.: Assessing the 'insider–outsider threat' duality in the context of the development of public–private partnerships delivering 'choice' in healthcare services: A sociomaterial critique. Information Security Technical Report 13(4), 202–206 (2008)
14. Orlikowski, W.J.: Sociomaterial practices: Exploring technology at work. Organization Studies 28(9), 1435–1448 (2007)
15. Latour, B.: Reassembling the social: an introduction to Actor–network theory. Oxford University Press, Oxford (2005)
16. Knorr-Cetina, K.: Sociality with Objects: Social Relations in Postsocial Knowledge Societies. Theory, Culture & Society 14(4), 1–30 (1997)
17. Bijker, W.E.: Of Bicycles, Bakelites, and Bulbs: Toward a Theory of Sociotechnical Change, 390 p. The MIT Press (1997)
18. Fosså, J.H., Mortensen, P.B., Furevik, D.M.: The deep-water coral Lphelia pertusa in Norwegian waters: distribution and fishery impact. Hydrobiologia 471, 1–12 (2002)
19. Beunza, D., Hardie, I., MacKenzie, D.: A Price is a Social Thing: Towards a Material Sociology of Arbitrage. Organization Studies 27(5), 721–745 (2006)
20. Björgvinsson, E., Ehn, P., Hillgren, P.-A.: Design things and design thinking: contemporary participatory design challenges. Design Issues 28(3), 101–116 (2012)
21. Suchman, L.: Human-Machine Reconfigurations: Plans and Situated Actions, 332 p. Cambridge University Press (2006)
22. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: writing a literature review. Mis Quarterly 26(2), XIII–XXIII (2002)
23. Levy, Y., Ellis, T.J.: A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. Informing Science Journal 6, 181–212 (2006)
24. Petersen, K., et al.: Systematic Mapping Studies in Software Engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE). University of Bari, Italy (2008)
25. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information and Software Technology 50, 833–859 (2008)
26. Oates, B.J.: Researching Information Systems and Computing. SAGE Publications Ltd. (2006)
27. Klein, H.K., Myers, M.D.: A Set of Principles for Conducting and Evaluating Interpretive Studies in Information Systems. Mis Quarterly 23(1), 67–94 (1999)
28. Monteiro, E., Almklov, P., Hepsø, V.: Living in a Sociomaterial World. In: Bhattacherjee, A., Fitzgerald, B. (eds.) Future of ICT Research. IFIP AICT, vol. 389, pp. 91–107. Springer, Heidelberg (2012)

29. Orlikowski, W.J., Iacono, C.S.: Research commentary: Desperately seeking the "IT" in IT research: A call to theorizing the IT artifact. Information System Research 12(2), 121–134 (2001)
30. Orlikowski, W.J.: The sociomateriality of organisational life: considering technology in management research. Cambridge Journal of Economics 34(1), 125–141 (2010)
31. Baptista, J., Newell, S., Currie, W.: Paradoxical effects of institutionalisation on the strategic awareness of technology in organisations. The Journal of Strategic Information Systems 19(3), 171–183 (2010)
32. Kuk, G., Davies, T.: The Roles of Agency and Artifacts in Assembling Open Data Complementarities. In: ICIS 2011 Proceedings (2011)
33. Orlikowski, W.J., Scott, S.V.: Sociomateriality: Challenging the Separation of Technology, Work and Organization. Academy of Management Annals 2, 433–474 (2008)
34. Riemer, K., Johnston, R.B.: Place-making: A Phenomenological Theory of Technology Appropriation. In: ICIS 2012 Proceedings (2012)
35. Johri, A.: Sociomaterial bricolage: The creation of location-spanning work practices by global software developers. Information and Software Technology 53(9), 955–968 (2011)
36. Gasson, S.: The Sociomateriality of Boundary-Spanning Enterprise IS Design. In: ICIS 2012 Proceedings (2012)
37. Thompson, T.L.: (Re/dis) assembling learning practices online with fluid objects and spaces. Studies in Continuing Education 34(3), 251–266 (2012)
38. Nicolini, D.: Zooming In and Out: Studying Practices by Switching Theoretical Lenses and Trailing Connections. Organization Studies 30(12), 1391–1418 (2009)
39. Al-Mahmood, R.: Spatial imaginings: Learning and identity in online environments. In: Markauskaite, L., Goodyear, P., Reimann, P. (eds.) Who's Learning? Whose Technology?, Proceedings, vol. 1, 2, pp. 43–54 (2006)
40. Law, J.: Notes on the theory of the actor-network: ordering, strategy, and heterogeneity. System Practice 5, 379–393 (1992)
41. Awazu, Y., Newell, S.: The Mangle of Practice in Enterprise System Implementation: Temporal Emergence and Material Knowing. In: ECIS 2010 Proceedings (2010)
42. Introna, L.D., Hayes, N.: On sociomaterial imbrications: What plagiarism detection systems reveal and why it matters. Information and Organization 21(2), 107–122 (2011)
43. Bratteteig, T., Verne, G.: Disentangling for Autonomy: Understanding the Sociomaterial Practices of Public Services. In: Keller, C., Wiberg, M., Ågerfalk, P.J., Eriksson Lundström, J.S.Z. (eds.) SCIS 2012. LNBIP, vol. 124, pp. 56–75. Springer, Heidelberg (2012)
44. Zorina, A.P., Avison, D.: When Environment Matters: Inter-Organizational Effects on Sociomaterial Imbrications and Change. In: ICIS 2011 Proceedings (2011)
45. Star, S.L., Ruhleder, K.: Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. Information System Research 7(1), 111–134 (1996)
46. Ciborra, C.: Formative contexts and information technology: Understanding the dynamics of innovation in organizations. Accounting, Management and Information Technologies 4(2), 61–86 (1994)
47. Mol, A.: The body multiple: ontology in medical practice. In: Science and Cultural Theory. Duke University Press, Durham (2002)
48. Leclerq, A., et al.: A Sociomaterial View of the Scaffolding of Work Practices with Information Technology. In: ICIS 2009 Proceedings (2009)
49. Monteiro, E., Rolland, K.H.: Trans-situated use of integrated information systems. European Journal of Information Systems 21(6), 608–620 (2012)

50. Monteiro, E., Jarulaitis, G., Hepsø, V.: The family resemblance of technologically mediated work practices. Information and Organization 22(3), 169–187 (2012)
51. Van Osch, W., Mendelson, O.: A Typology of Affordances: Untangling Sociomaterial Interactions through Video Analysis. In: ICIS 2011 Proceedings (2011)
52. Orlikowski, W.J.: Material knowing: the scaffolding of human knowledgeability. European Journal of Information Systems 15(5), 460 (2006)
53. Østerlie, T., Almklov, P.G., Hepsø, V.: Dual materiality and knowing in petroleum production. Information and Organization 22(2), 85–105 (2012)
54. O'Farrell, J., Letch, N., Purchase, S.: ERP Usability and the Mangle of Practice: Improving Work Processes and Employee Performance. In: AMCIS 2012 Proceedings (2012)
55. Svahn, F., Henfridsson, O., Yoo, Y.: A Threesome Dance of Agency: Mangling the Sociomateriality of Technological Regimes in Digital Innovation. In: ICIS 2009 Proceedings (2009)
56. Scott, S.V., Orlikowski, W.J.: Reconfiguring relations of accountability: Materialization of social media in the travel sector. Accounting, Organizations and Society 37(1), 26–40 (2012)
57. Nardi, B.A.: Beyond Bandwidth: Dimensions of Connection in Interpersonal Communication. Computer Supported Cooperative Work (CSCW) 2(14), 91–130 (2005)
58. Ribes, D., Finholt, T.A.: The Long Now of Technology Infrastructure: Articulating Tensions in Development. Journal of the Association for Information Systems (JAIS) 10(5), 375–398 (2009)
59. Karasti, H., Baker, K.S., Millerand, F.: Infrastructure Time: Long-term Matters in Collaborative Development. Computer Supported Collaborative Work (CSCW) 19(3-4), 377–415 (2010)
60. Latour, B.: The Politics of Nature. Harvard University Press, Cambridge (2004)

Author Index