Aalborg Universitet

Apathy towards the Integration of Usability Work

Wale-Kolade, Adeola; Nielsen, Peter Axel

Apathy towards the Integration of Usability Work: A Case of System Justification

Adeola Wale-Kolade

Department of Information Systems

University of Agder

Norway

Peter Axel Nielsen

Research Centre for Socio+Interactive Design

Department of Computer Science

Aalborg University

Denmark

Abstract: In this article we report from a case study of a software development organization and we study in particular the developers' and product managers' attitudes towards integrating usability work into software development. We offer explanations based on system justification theory illuminating what would-be integrators might be up against. The analysis shows how the developers only pay lip service to usability work and how they treat users superficially. It further shows how that leads to stereotyping of usability designers and users in order to preserve status quo, and how internalization of inequality between the developers and usability designers rationalizes the preservation of status quo. These findings will have significant implications for managers of software development.

Research Highlights

- Software developers may claim to adhere to the principles of usability design and evaluation while their practices show the contrary

- Usability designers and users may become stereotyped and internalise a perception of inequality.

- Developers may cling to status quo and seek to preserve it.

- Managers of software development and usability work must pay attention to justification mechanisms to realise the challenges of integration.

1. Introduction

A software development company flashes the following text on the front page of their web site: "We help you with user-friendly solutions …" It was contradictory then to learn later in an investigation of the company that the company has little interest in usability design and evaluation, let alone in their users' experience of usability. It is on that backdrop that we went further in our investigation to try to figure out why that is. That is the crux of the matter of this article: Why?

Over the last decade, there has been marked improvement in the software development industry's perception of usability work (Chilana, Wobbrock et al. 2010). Previously thought of as a vague and immeasurable system attribute, usability is now being acknowledged within the industry as a means through which companies can maintain their competitive edge and customer base (Mao, Vredenburg et al. 2005). There is also now greater acknowledgement of the usability designer role as a valid profession (Gulliksen, Boivie et al. 2006), and more individuals with this skill are involved in development projects (Venturi and Troost 2004).

For a field that has struggled for legitimacy since its establishment (Rajanen and Iivari 2007), these positive developments would seem to suggest that usability work has achieved considerable integration, such that it is now a common consideration in software development projects. Findings from practice paint a more complex picture; while there has been some progress, it is not as widespread as is to be expected (Gulliksen, Boivie et al. 2006; Ji and Yun 2006; Venturi, Troost et al. 2006; Abrahão, Juristo et al. 2010). Indeed, it seems to have had no discernible impact as most organizations still lack a strategic approach to usability work (Seffah and Metzker 2004; Zhou, Huang et al. 2008; Ardito, Buono et al. 2013). Practices such as a focus on the user and end-to-end user involvement are yet to become the norm (Mao, Vredenburg et al. 2005). Usability designers still do not have the clout they need to function, and they often end up being passive bystanders during software development projects (Gulliksen et al. 2006). Generally, there a gap between intention and reality in the software industry remains (Bygstad, Ghinea et al. 2008).

In this paper we seek to understand the reasons behind this and therefore ask, "How can we explain the reluctance in practice to integrate usability work into software development?" We conducted an interpretive case study of an agile software development company to examine the attitudes towards the integration of usability work. In accordance with previous studies of the integration of usability work into software development (Bygstad et al. 2008; Iivari 2006; Gulliksen et al. 2003; Boivie et al. 2006) we have taken a process view rather than a product view. To provide a rich picture of participants' rationalisations on this matter, we applied the theoretical lens provided by system justification theory. The theory is one which builds on individuals' natural tendency to rationalize and how they use such rationalisations to justify the social systems they exist in – even at the expense of personal and group interest (Jost et al. 2004).

In the next section we provide an overview of existing research on the integration of usability work within the software development industry. In section 3 we describe system justification theory as our theoretical lens. Section 4 argues for the research design, the data collection and analysis, and it provides an overview of the selected case organisation. Section 5 presents our findings where we show the system justification mechanisms being engaged in the case organisation. We discuss the findings in section 6 and relate these to the existing literature.

2. Related research

In this section we provide an overview of existing research on the integration of usability work.

2.1. Usability work integration

Usability work consists of activities such as usability design and usability evaluation carried out to ensure a desired level of usability in the final product. There is an interplay between these activities evident in how

products from the design phase form the basis for the evaluation phase; and how findings from the evaluation phase are fed back into the usability design phase for subsequent development of the product (Hornbaek and Stage 2006). Usability work is said to be integrated when there is a strategic decision to fully incorporate it into an organization's business processes (Venturi, Troost et al. 2006). When this occurs, the following six attributes can be expected to characterise such an organization (Venturi, Troost et al. 2006). These are:

- Usability work would be included in a timely way into the product lifecycle

- The usability team would have access to proper skills and experience

- There would be a proper usability infrastructure in place to support the usability team

- Management would be fully committed to usability work and to ensuring it is done

- Usability work awareness and culture would be properly disseminated inside and outside of the organization

- Results from usability work activities would inform design decisions.

Another approach is the usability maturity model (Nielsen 2006) that can be used to assess the integration of usability work in an organization. The model describes an eight-level evolutionary path which shows the stages an organization progresses through as its approach towards usability matures, see Table 1.

<<Insert table 1. Caption: Usability maturity model (Nielsen 2006)>>

2.2. The integration of usability work in practice

Several studies (Venturi and Troost 2004; Mao, Vredenburg et al. 2005; Venturi, Troost et al. 2006; Bygstad, Ghinea et al. 2008; Ardito, Buono et al. 2013) have examined the integration of usability work within the software industry. Their main finding is that even though there have been marked improvements in organizations' awareness of usability work and its importance, organizations where usability work is fully integrated are still rare (Venturi, Troost et al. 2006; Bygstad, Ghinea et al. 2008; Ardito, Buono et al. 2013). More specific observations include: little or no user involvement in software development projects (Vredenburg, Mao et al. 2002; Vukelja, Müller et al. 2007); usability issues being handled by software developers despite their limited understanding of usability work (Gulliksen, Boivie et al. 2004; Vukelja, Müller et al. 2007); and the disparity between their mental models and that of users (Høegh and Jensen 2008). Hence, usability work is ad hoc and unsystematic and there is a wide gap between what organizations acknowledge usability work to be and what they actually do (Venturi and Troost 2004; Ji and Yun 2006; Vukelja, Müller et al. 2007; Bygstad, Ghinea et al. 2008; Zhou, Huang et al. 2008).

Some researchers have attributed the non-integration of usability work to a lack of education among software developers and management (Vukelja, Müller et al. 2007). It has also been observed that organizations are unwilling to expend resources on usability work (Bygstad, Ghinea et al. 2008) because of ambivalence about benefits for the organizations (Mao, Vredenburg et al. 2005; Bygstad, Ghinea et al. 2008; Zhou, Huang et al. 2008). Cost benefit trade-offs have been highlighted as a reason why organizations prefer to use heuristic evaluations rather than the more expensive field study techniques (Vredenburg, Mao et al. 2002). Organizational cultures have been studied and particular cultural characteristics are more suited to certain types of usability work (Iivari 2006). Other reasons are that usability is a non-issue among clients as these

individuals do not specifically request for it when initiating a project (Ardito, Buono et al. 2013). Time pressures in software development projects have been noted (Boivie, Åborg et al. 2003). There is also the issue of the type of development method, which is a criticism that has been levelled against agile methods in particular (Blomkvist 2005; Chamberlain, Sharp et al. 2006).

Solutions to usability work integration are dominated by methods and prescriptive approaches in the literature and these have been criticised for failing to acknowledge the reality of practice (Gulliksen, Boivie et al. 2006). In some cases these solutions have even been observed to work to the detriment of usability work. A case in point is the usability cost benefit model which has been highlighted as being inherently capitalist in orientation and encouraging focus on the paying customer and not the end user (Rajanen and Iivari 2007). One approach which is increasingly gaining ground in the usability work integration debate concerns the organizational character (Cajander, Gulliksen et al. 2006). Obstacles to the integration of usability work within software development are a consequence of how organizations perceive usability work and if not conducive, integration becomes impossible (Gulliksen, Boivie et al. 2006). In this regard it has been shown that managers have a structural perspective where the emphasis is on efficiency and economy and that this explains their viewing of usability work as a non-rational investment, even if they acknowledge its relevance (Cajander, Gulliksen et al. 2006).

Agile software development has received much interest among practitioners over the last years, and has led many software development companies to adopt to agile principles, e.g., a focus on flexibility and amenability to changes in business requirements through iterative processes (Dybå and Dingsøyr 2008). This in turn has led to much research interest into how usability work is achieved within agile settings; and it has been observed that agile processes present non-trivial constraints for usability work (Ferreira, Sharp et al. 2011). Research into the integration of usability work with agile software development has just started and most results are normative so far with little empirical investigation or detail (Wale-Kolade, Nielsen et al. 2013). Both agile development and usability work are similar in many respects, e.g., iterative approach, user involvement, and prototyping (Blomkvist 2005), and several attempts at combining the approaches have however led to little integration (Lárusdóttir, Cajander et al. 2012). The integration seems to be problematic due to several factors, e.g., power struggles between software developers and usability designers (Chamberlain, Sharp et al. 2006), and difficulties with separating customers from users (Blomkvist 2005). These difficulties can be overcome (Kautz 2011), but there are very few reported successes (Ardito, Buono et al. 2013).

3. System Justification Theory

System justification theory was developed to provide insight into how and why individuals legitimize and support their social systems even when these systems embody values that conflict with their own self-interest (Jost, Kivetz et al. 2005), and even when made aware that a different social system might better meet their needs (Napier, Mandisodza et al. 2006). It asks why individuals reinforce prejudicial attitudes that exist within their social systems.

Here social systems refer to the structured network of social relations which may be tangible such as the families, institutions and organizations within which people live their daily lives or abstract and intangible, such as the unwritten but clearly recognizable rules and norms that prescribe appropriate social behaviour

(Thorisdottir, Jost et al. 2009). With regards to the theory, it has been asserted that if individuals' blind adherence to the status quo is to be overcome and if interventions are to be effective, then there is a need to understand the rationale for their actions (Haines and Jost 2000). This is as ignoring these self-justifying tendencies would only lead to errors in the prediction of individuals' attitudes and inadvertently the solutions proposed to tackle them (Blasi and Jost 2006). The disconnect between the existing recommendations on the integration of usability work and practice and their inability to resolve the usability problem in software development, cf. (Ardito, Buono et al. 2013), may be seen as in accordance with this argument put forward by the theory. Further the priority the theory places on understanding individuals' attitudes and motivations is a practice which in the usability literature has been described as necessary to change organizations' approach to usability work (Cajander, Gulliksen et al. 2006; Gulliksen, Boivie et al. 2006). This shows the theory's relevance to our study.

The theory asserts that individuals in engaging in system justification are driven by their social and psychological needs to legitimize the status quo and see it as good, fair and inevitable (Jost, Banaji et al. 2004). It is this emphasis on the motivational component of individuals' defence of the status quo that distinguishes system justification theory from status-quo-biased theory. The latter being a theory that considers such bias a consequence of a purely cognitive process involving the consideration of factors like transaction costs during choice making (Kay, Gaucher et al. 2009). Conversely in system justification theory this bias presents itself as a paradox, manifesting in individuals' legitimation of their social systems even when aware of the shortcomings of their social systems (Gaucher and Jost 2011).

System justification theory describes the mechanisms that might be employed by individuals when justifying their social systems (Jost, Banaji et al. 2004; Jost, Kivetz et al. 2005; Blasi and Jost 2006). They include in no particular order, stereotyping, internalization of structural inequality, and rationalization of the status quo:

- **Stereotyping:** Describes how individuals use mental representations of social categories within a system to defend disparities in how individuals within a system are treated (Kay and Jost 2003) and system passivity towards these disparities (Napier, Mandisodza et al. 2006). An example is the 'poor and lazy' vs. 'rich and hardworking' stereotype which leads individuals to justify why the poor should be neglected by the system and the rich esteemed (Jost, Kivetz et al. 2005; Singh, Mathiassen et al. 2010). Another form of stereotyping is complementary stereotyping where compensating favourable virtues are ascribed to a disadvantaged category (e.g., 'poor but happy') and corresponding unfavourable vices to the advantaged category (e.g., 'rich but sad') (Kay and Jost 2003).This latter form of stereotyping has been observed to allow individuals to maintain a psychological sense of equality ('no one has it all'), and justify inequality ('people get what they deserve and deserve what they get') (Jost, Kivetz et al. 2005; Kay, Jost et al. 2005).

- **Internalization of structural inequalities:** Describes how disadvantaged individuals within a system legitimate the manner in which they are treated by the system in a bid to maintain their view of the system as fair and just (Blasi and Jost, 2006; Napier et al. 2006). It may in some cases involve some bias in information processing such that these individuals construe the situation in a manner that portrays them as deserving of whatever unfairness they experience (Napier et al., 2006). It may also be present as implicit or explicit group favouritism whereby members of disadvantaged groups

evaluate advantaged groups more favourably than they evaluate themselves (Gaucher and Jost 2011). While this occurrence lessens whatever emotional distress the individual might experience, it leads to a 'depressed-entitlement' effect (Blasi and Jost, 2006). This is a condition where such individuals out of the continued persistence of the issue become completely resigned to the current situation and feel they do not deserve better from the system (Napier et al., 2006). Thus they become less inclined to advocate for change.

- **Rationalisation of the status quo:** Describes how individuals justify not only their own decisions and behaviours but also the decisions and behaviour of others as well as outcomes where no decision maker was involved (Blasi and Jost, 2006). These justifications may involve denying or ignoring system failures in a bid to avoid the negative emotions that would result from admitting to these system failures (Napier et al. 2006). Additionally they might present as 'sour grapes' rationalizations whereby objects perceived as unavailable or unattainable are disparaged; and as 'sweet lemons' rationalizations where emphasis is placed on the desirable features of present unimpressive attainments (Kay, Jimenez et al. 2002; Sinha 2013).

System justification theory further highlights that varying situational and dispositional conditions may cause the specific justifications and how they are triggered. This enables explanation of why in some contexts or among certain individuals, the tendency to engage in system justification is stronger in some situations and weaker in others (Kay and Friesen 2011). Criticisms by outsiders have been identified as the most potent form of the system threat antecedent (Kay and Friesen 2011).

4. Research Method

The research is based on the case study approach (Benbasat, Goldstein et al. 1987; Lazar, Feng et al. 2010; Yin 2013) to investigate a contemporary phenomenon in depth and within its real-life context. Both within research on software development and on usability work, the use of the case study approach has been advocated because it is more suited to the dynamic and complex nature of the field (Zhang, Li et al. 2009). The case study approach is particularly appropriate as the research question addresses the 'how' and 'why' of the integration of usability work into software development.

In designing the case study we adopted the single case strategy in line with the common case rationale (Yin 2013). The aim is to capture the circumstances and conditions of an everyday situation to gain insight into the social processes surrounding the phenomena being examined. In addition we have assumed an interpretive stance (Walsham 1995) with an added inclination to probing beyond surface level explanations (Miles and Huberman 1994; Myers and Newman 2007).

The case company was selected because the company at one level portrays itself as delivering software products and services with a high level of usability and at the same time performing agile software development. The software company is small with approx. 30 employees and as such typical for software companies in Scandinavia.

Six informants were selected including the key product manager, see Table 2. In order to avoid elite bias (Myers and Newman 2007), the informants were selected based on their involvement in product development, rather than their position in the organization. The interviews were semi-structured following

general advice on qualitative interviewing (Patton 2005; Myers and Newman 2007) and utilizing an interview guide.

<<Insert table 2. Caption: Profile of informants>>

The interview guide framed the interviews in a semi-structured format, asking the participants questions related to: (1) their awareness of usability work, and (2) how usability work should be integrated into software development in general and agile development in particular. The interviewees' awareness of usability work was measured in terms of articulated descriptions of usability and experience with usability training. Questions relating to how agile methods and Scrum in particular were used in the company were also asked to provide context to the study. The qualitative interviewing was conducted in between an hour and an hour and a half for each interviewee. The interviews were audio recorded. In the post processing the interviewer also contacted the interviewee to clarify some of their answers.

The analysis was done using system justification theory as the theoretical lens (cf. section 3) and applying contents analysis (Lazar, Feng et al. 2010). The interviews were first read to gain an overview and form an initial structure and interpretation. Subsequently, the interviews were analysed in detail by identifying the quotations interpreted as relevant for the topic and then coding these quotations using the categories from the system justification theory. Finally, the quotations and their coding were reviewed and reconsidered to ensure that the coding was consistent and in accordance with the theory. The interviewees' expressions and tone of voice were also taken into account during the analysis.

We have in the analysis applied a process view rather than a product view (Floyd 1987). This means we have a particular interest in how they work, how they develop software, and how well they integrate usability work. This view is in line with previous studies of the integration of usability work into development (Bygstad et al. 2008; Iivari 2006; Gulliksen et al. 2003; Boivie et al. 2006), and this is also the focus chosen in research on integrative methods, e.g., (Göransson et al. 2003; Gould & Lewis 1985). In a similar manner we have analysed our case data with a view to the process questions such as 'in which ways?', 'how well?', and 'why not?'

5. Analysis and Findings

In this section we first introduce the case company and provide an overview of how they develop software; we then present the findings.

5.1. The Case Company

The case company is located in Scandinavia and started out as a software development company more than ten years ago. A few years ago the case company was purchased by a leading international company within enterprise systems. The primary task of the case company is to develop software. It operates as an independent subsidiary of the parent company, and it has a high level of autonomy and control over its operations. The product suite consists of web-based applications for processing projects, time, travel and expenses. The company has long-term relationships with a few large customers and its products are not aimed at a mass market. The case company has a forum of super users that is established prior to customers

purchasing the product and is active after deploying the product. The super users' forum acts as a liaison between the software company and its customers. These super users are typically the IT personnel in the customers' organizations and they are also responsible for selecting the configurations of the product suite. On their web site it says: "We help you with user-friendly solutions …" Usability of the product suite is not only a sales tactic, it is also a business goal.

The company is small and staffed with 29 employees. The software company's product development team consists of nine people. Six of these are developers, one is a functional tester, one is the product manager, who acts both as a Scrum master and a product owner, and one is a lead consultant. The work environment is open and encourages collaboration, sharing of information, and a large degree of autonomy for developers. The company has a customer centre that provides consultancy, support, training, and implementation services directly to customers. The customer centre also provides help desk service to the users.

5.2. Agile development in the case company

The software company has for the past five years been using their own, adapted version of Scrum (Schwaber and Beedle 2002). Their version of Scrum is modified with respect to how they organise their daily stand-up meetings and sprint planning. For instance they have two daily stand-up meetings unlike the one that is suggested by Scrum. The first meeting is between the product manager and the lead consultant and it lasts no longer than 15 minutes. According to the informants, this arrangement was introduced to further the communication efficiency between the developers and the consultants by: (1) creating a single gateway through the product manager, and (2) preventing consultants from overloading developers with requests. Later in the day there is a standard stand-up meeting for the product development team lasting no longer than 15 minutes and aiming at coordinating the work of team members.

At the start of a new iteration sprint the team spends half a day on sprint planning. Sprint planning includes which tasks should be completed, but it excludes effort estimation based on hours. Rather they plan based on story points. The interview with the product manager reveals:

> "Earlier we use[d] to plan each task with hours on each developer during the sprint planning and that means if one developer has 250 hours we will break that down and he will take on the tasks that he wants to have from the list and suggest some hours on that. What we have done recently since April this year has been a shift toward that we have a pool of tasks and we have not planning on hours."

The product manager who is also acting as Scrum master monitors the time spent by developers on tasks in order to minimize delays in subsequent, dependent tasks.

The team has 14-days-sprints and maintains production line setting where the scope of each sprint is not fixed. Sometimes the outcome of a sprint is bug fixes on a version of products or product upgrades. Sometimes the outcome is new functionality specifically requested by key customers or newly acquired customers. Sprint reviews occur sometimes at the end of the sprints.

5.3. Usability work in the company

The informants' responses show a reasonable level of knowledge of usability work. Themes emerging concern users' interaction with the system, learnability, and usability. This is captured in several statements, e.g., the product manager states:

> "I would describe it as the user interface and paying attention to it would be making the software as good as possible for the user to use. So it is not only about the functional part. One thing is that it has to be functional but it also should be a good experience to use it and that is the layer that I would put into usability."

One of the developers states that usability is about having the "user perspective" and highlights the importance, as a developer, of having an outside perspective as "sometimes if you are working on a product, you work every day with the same details, you get kind of blind in your ideas."

The informants are also able to describe various ideal usability work practices and their perceived benefits, see Table 3.

<<Insert table 3. Caption: Evidence of the informants' knowledge of integration of usability work into their agile software development>>

The informants' understanding of usability work is not limited to their ability to provide articulate descriptions. The functional tester has previously been employed as a usability designer in another company. The product manager has also taken courses in usability while at university and is generally familiar with usability design principles.

This background knowledge on usability work is a prerequisite for the business goal to pursue a high level of usability. It is however worth noticing that on the one hand they acknowledge that the integration of usability work could be valuable to them, but on the other hand they do not integrate usability work into their development process.

The current development practices in the case company are not integrating usability work. One of the developers said:

> "We don't actually really speak to the end users."

On the question of whether they do usability testing with users the product manager replies:

> "... no I wouldn't say so. The only way we have touched that would be when we did the expense overview, but that was also much based on functionality."

One of the developers describes how a usability evaluation session is carried out:

> "... we hired a graphic designer, we did not do much testing just some testing to verify with our biggest customers that the new icons and colours were okay."

Apart from the functional tester, the other members of the product development team do not feel that the integration of usability work into their development processes is something that needs to be approached more systematically. These team members describe themselves as user-driven, asserting that their products "should be self-describing." According to the lead consultant, it is "very important to us to have a system which is good for the users." This again confirms the business-oriented usability goal of the case company. Yet the team never conducts an evaluation of the usability of the products. The development team collects

users' feedback; but that concerns solely perceptions of the company and the functionality offered in the product suite.

The developers, the lead consultant and the product manager are aware of the inherent and potential danger in this approach, but they have chosen to accept this. The product manager asserts that it is possible that their product "could be really a [poor] system and we are just surfing around happy here, thinking everyone loves our system and they are really like [not]." This shows a discrepancy between the desire to develop for high usability and how systematic the manager pursues it.

The functional tester, who is also a trained usability designer, further states:

> "It is funny because at this user conference, we showed our users a really small change which was only one line of code and it was that when you click on expense print out, you get a view of what it would look like. Before you had to use the [image] of the expense and then you had to click print once more and then you had to click on the printing interface once more. The company got applause for this and everyone was shocked. It took the developer maybe a minute to do that and they got applause. It was a good example of usability, that only small things can get our customers to go 'wow.'"

The quote shows that the developers have no practical awareness of usability work since they have not tested its usability with their users and they have little sense of the ease with which they could potentially increase usability.

The company has no infrastructure in place for usability work. There is no usability designer role in the product development team. It is likely that the company is too small to have a full-time usability designer, but there is none assigned to a part-time role either. The functional tester with the usability designer skills and who might have functioned in a part-time role as usability designer has been assigned a different role and to different tasks. Management is not committed to a process that includes usability work as captured in the following comment by the product manager:

> "So I would say that we consider [usability work], but we don't do it, we don't have like this is the usability way ... this is the design team, this is the development team, we don't do that, but it is always there. It is more implicit than explicit, we would never release anything without having checked that it works with the client, but how we do it is not set and sometimes we do it too late."

The reluctance to integrate usability work in an explicit way is not a consequence of their agile development approach or specifically their use of Scrum. As emphatically stated by one of the developers, "No it nothing to do with Scrum. ... it is nothing to do with which project methodology you use, it is the decision you as a company take." This view is also shared by the functional tester who asserts, "No. Actually I think it [Scrum] is really good for usability, because we can do some rounds and think about it. ... We have the possibility to do it, if everybody agrees that now this is necessary." According to the developers, Scrum is a flexible approach as it allows for more "change than in waterfall" and that one can "test things in an earlier stage using Scrum than in waterfall."

The company's approach to usability design seems independent of their agile development approach, but there is no deliberate integration of usability work into the development process. The company as summed up by the functional tester is "missing a routine" to usability work even though it "has the tools", and she and

others have the knowledge needed. This resembles the level of developer-centric usability in the Nielsen model, cf. Section 2. The product development team realises the value of usability, but responsibility for ensuring the usability of their products has been left to the developers, who on their part are based on intuition and an understanding of what is efficient.

The analysis shows that the company can best be described as paying some kind of lip service towards usability work. They claim usability work to be important, but they do not practice it in a systematic or explicit way. They can talk about it and they can explain the different aspects in detail including its potential benefits. They have however deliberately chosen not to pay more attention to it.

5.4. A system justification analysis

The analysis of the development team's rationalizations in defence of their approach to usability work can be extended by applying the theoretical concepts of system justification theory. The following analysis shows the three mechanisms we found in the interviews: stereotyping, internalization of inequality and the rationalization of the status quo. Stereotyping captures the descriptive terms participants used when talking about the usability designer's role and their users. The internalization of inequality captures the viewpoint of the usability designer and how this individual now in the functional tester role rationalised the approach of the organization. The rationalization of the status quo includes two different aspects both leading to the preservation of status quo: justifications that highlight perceived impracticality of usability work, and justifications that highlight the company's present achievements.

5.4.1. Stereotyping

The stereotyping addresses both usability designers and users.

Usability designers as creative, but orthodox

Stereotyping is observed in the development team's descriptions of usability designers and the approach the company has adopted towards this particular role. For instance the developers recognize the benefits a usability designer would bring as highlighted in Table 2. Further terms such as "creative," "fun to work with," and "particular" were used by the developers to describe some of their impressions of usability designers. However, the product manager argued at the same time that usability designers were often out of touch with reality and users' needs and that they had failed to recognize that the "rules had changed within usability." Further that:

> "designers think in more general terms, you know less is more, sometimes they talk about the audience, but they forget that it is a bit different... Of course it is fun to have an application that looks cool, but the most important thing is that it runs fast."

This line of rationalization came as a response to a question of why the company do not employ a usability designer (perhaps part-time) in the light of usability being one of the business goals. From a business perspective it may well be reasonable not to employ a usability designer and then in stead rely in the software developers to take responsibility for usability design. From the product manager's defensive attitude we suggest an alternative interpretation that is part of a stereotyping of usability designers.

The above statements also came in a discussion concerning an aspect of their products' interface which apparently has been criticized by the usability designers they have met over the years, on the basis of it violating usability design principles. The product manager responded rhetorically, "should we listen to the professional guys or should we listen to our clients?" As we shall see below they have a process where they listen to super users, but not to all their users.

The developers also mentioned that usability designers ignore that the "demands for usability were different when it comes to administrative systems" and how the users of such systems are more focused on speed rather than whether the application is "nice and colourful." The developers mentioned that their clients had affirmed that that is the main reason they choose the company's products. The developers view usability designers as individuals who are out of touch with reality and are too focused on the mundane or even orthodox. For example, one of the developers when asked about the impact a usability designer has had on an application that was redesigned, replies that it is more "glossy and colourful." This is effectively a very limited view on usability and usability work. The developers generally rationalised that having a usability designer would lead to unnecessary interface changes.

We are not suggesting that this stereotyping of usability designers is the main issue when the product manager and the developers justify a discrepancy between the business goals and the process through which they work with usability; but we are suggesting that it is a part of it.

Super users as central gatekeepers; end users as futile

The product development team distinguishes between 'super users' and 'end users.' The super users perform the dual role of acting as a gateway and gatekeeper between the customer organization and the software company. These super users act as primary internal support persons in the customers' organisations and are typically employed in the information technology department or financial department of the client organization. According to the developers the super users are the "power users who use all of the system and not the normal users" whose use of the system is limited to the occasional "delivery of hours and recording of expenses."

The company place a high priority on the super users. According to the product manager, this is because they are the "contact point" and therefore it is not "natural" to relate with the end users. According to one developer, the super users are "experienced so small glitches or something that could be done easier does not get their focus, the focus is more on the functionality and not 'can we do this in this way?'" The product manager mentioned that if the company decides to make any changes to the user interface, with a focus on making it easier to use, "then we would [need to have] good arguments for doing that and we would need to train them [super users] in using the interface." In other words, they would need to justify why they would want to improve the usability of their products, as in this case for the super users, it would be a "hassle."

The developers also argue that it is not necessary for them to focus on the end users since their products are "for companies anyway, for enterprises; [and] not for end users." To explain the incongruence in this statement, one of the developers stated, "it is for end users, but not primarily for end users and in that case, it is more functionality that is important than if it is green or red." In other words, the developers feel the end

users are incapable of knowing what attributes a good product should possess or making judgements related to it.

Additionally, the developers acknowledged that there are differences in the tasks these two sets of users carried out on the system, but they do not perceive that it is necessary to examine the use experience of the end users. The head consultant asserts that the registration of hours on the system that the end users carried out is typically a "pain … for most people" and their aim is that "it should go very easy and that is why we have to focus on user friendliness." This consultant talking for the product development team said they do not want a situation where the end users will have the "thought after they have used or [entered in] expense, this is awful and I don't want to do it again." In spite of these affirmations, the development team takes the perspective of the super users and even when the focus is on functionality. Asking the head consultant about this, highlighting the incongruence, he again justifies their focus on the super users, stating that it is important "they [super users] should know all the functionality and they can select what is the best for [their] company."

The developers rationalise that these super users should be given high priority in the development process. This priority is in the form of courses arranged for these users, occasional testing of new functionality with the super users, and eliciting of company-related feedback from these users. For the end users, it should be enough that they follow the help guidelines the company has installed in the products.

5.4.2. Internalization of inequality

In examining the internalization of inequality we focus on the functional tester who is a trained usability designer and who is most disadvantaged by the current arrangements within the company. The frustration this individual feels can be gathered from:

> "I work as a functional tester, this is my job and they knew that I had a background for usability,
> but I have no percentage, 10% or 20% of my time I can focus on usability"

The functional tester admits to the failures of the company in relation to integrating usability work into the development process. She mentions how the user perspective has been neglected; and she refers to how the focus is on the super users and not on the end users the few times usability testing has been performed. This, she notes, has negative implications as the super users are using other parts of the system than the end users. She also elaborates more on some of the occurrences that has been used by the developers as a basis for their justifications, e.g., user feedback was positive only because "it is typical Scandinavian to give good feedback" because "the salesmen had done a good job of advertising the product as one with good usability," and because the company mainly obtained user feedback from the highly prioritized super users. She further states that the "general understanding" among the developers is that having decided on the right colours solves most usability issues. For her usability is that:

> "a specific user can solve his problem. The user interface can have something to do with
> usability, but not necessarily I think. I think [the user interface] is just the tool. It could also be
> work flows or something like this … But it is … not only the user interface, but I understand the
> developers think it is."

She also described how she has once presented at a seminar to argue for more focus on usability in the development process, and she also suggested that they try usability testing during sprints for a month. However, such considerations were not acknowledged. This failure to make any changes, it seems, has led to her becoming gradually more accepting of the way things are. This is captured in the following:

> "I am really interested in this [usability work] and I would like to do it more. But I know it is not my main job and I have a lot of things to do in my functional testing and so on. That is how it is."

As a sign of this gradual shift towards acceptance, we observed that the functional tester sometimes legitimates current practices in the organization. For instance when asked why no usability testing is being done, she mentioned that it is "because we are a small organization, and often we do not have time." She understood the position of the product manager since there are "a lot of tasks to do and customers want new releases all the time and bugs fixed really fast." She also talks about how she thinks the "developers have a focus on usability," and "that they have it in their head," thereby contradicting her earlier assertion on the developers' lack of interest in usability work as part of the process.

The functional tester also lacks experience with usability work in agile development processes. She has emphasized the need for usability testing to be conducted in between sprints. However, when asked how easy this was to do in her previous place of employment, she admitted these suggestions were mostly in her "head" and that she had never seen it done in practice. However, she do not see her "disadvantaged state" as being a consequence of the company's use of agile processes. This is summed up in the following:

> "I really like the agile way of working and I am part of the team and I don't have to spend as much time discussing or arguing."

The balance of power is clearly not in functional tester's favour, yet through the internalization of inequality she also in part rationalizes the organization's approach to usability work.

5.4.3. Rationalisation of status quo

The rationalization of status quo is twofold that usability work is not worth integrating into the development processes ('sour grapes'), and that the developers are already doing everything right ('sweet lemons').

'Sour grapes' rationalizations of usability work

Usability work is described by the developers as "more theoretical than practical." Though the developers have provided descriptions of how usability work should ideally be done (cf. Table 3), one of the developers argues that "in our project it is not a realistic scenario" to perform usability work like user trials in their projects. Usability testing is considered to be something necessary only when it would give a direct effect and not something to be done systematically. One developer describes testing as an impossible feat as it is "difficult to know who your next user would be," even though he admits that this could easily be resolved by taking "anyone from the street and testing the user interface." The developers also talk about the cost of testing, in terms of the manpower and analysis involved. Other concerns include time in relation to how actions like prototyping during sprints and having these tested with users would lead to project delay. The product manager remarks, "I think in a perfect world where we did not have to think so much about how much time we use when we are doing things, we could focus on usability."

Employing a usability designer (even part-time) as part of the development team is not seriously considered, even though they acknowledge that it might be beneficial. One of the developers talked about how such an

individual as part of the team "could get caught up in the same mind-set, the same ideas, [and] the same pattern of doing things" and that it is preferable for such an individual to be outside the organization so as "to have an unbiased view of the product." This developer also argues that the company is not "big enough a company or organization to be able to have someone with that sole responsibility."

Thus, the potential benefits of usability work are diminished – what could potentially be sweet grapes are turned sour.

'Sweet lemons' rationalizations of self

One of the developers argues that, "we try to do a reasonably good job, so we try to make it workable... we do compromise on some things sometimes, we don't go by the book but till now it seems as if the people who use our product, we get good feedback".

This feedback they describe as including clients' favourable comparisons of the company's products against those made by their competitors, high number of business referrals, and low customer attrition rates. This positive feedback is used to extol the benefits of the company's current development process. For example, the product manager states:

> "Our users are giving us high credit even though our applications defy usability design guidelines and it is because we focus on functionality."

The developers also emphasize the altruistic aspects of their development process. One of the developers highlights how they have included the functional layer into the graphical user interface to make the applications more responsive for the users. Another developer in relation to this admits this has limited their ability to make changes to the interface. However, this is again justified on the basis that it allows them to ensure that customers' familiarity with the products is maintained. But a developer states that "we have solved this reasonably. Most of our functionality is user friendly."

Thus, the developers contend that they have good development practices and that this has been proven by the lack of customer complaints. That helps inflating the belief in what they are currently doing and in preserving status quo.

6. Discussion

At a distance the case data may lead us to consider it along the same lines as previous research which has examined the gap between the research and practice and concluded how prescribed or formalized development methods are rarely adopted in their entirety and never used as prescribed (Stolterman 1991; Fitzgerald 1998; Nandhakumar and Avison 1999; Fitzgerald, Russo et al. 2002). Previous research has also highlighted how such adaptation of methods and processes are often based on pragmatic considerations (Stolterman 1991; Fitzgerald, Russo et al. 2002). At one level of analysis our findings concur with this as we have also found discrepancies between research and practice, e.g., the development team has made modifications in their practice of Scrum as an agile method.

The system justification theory as an analytical lens has led to a closer look at the case data, and at a qualitative explanation of the case company's contradictory stances towards the integration of usability work into their software development processes. The analysis has shown that not only do contradictory stances

exist, but the analysis also suggests the source of this. There are several contributions in the analysis and the findings from the case data.

Firstly, the analysis based on system justification theory tells us that sometimes such adaptations of methods and processes with respect to usability work may well arise from irrational considerations such as a preference for the status quo and apathy towards change. The theory also enabled us highlight the rationalizations which could be used to legitimate such bias. In this study, this referred to the development team's stereotypical representations of users and usability designers, the designer's internalization of inequality. Lastly the 'sour grapes' rationalizations where the developers saw usability work as impractical; and their 'sweet lemon' rationalizations of self which bolstered their confidence in their approach. There was in general a self-reinforcing cycle which ensured a continued sustenance of the apathy towards usability work in the organization and which ensured there would be no integration occurring in the organization.

Secondly, the findings from the case highlight unanticipated attitudes - in this case the tendency to engage in system justification and the rationalizations which accompany this. Such unanticipated attitudes have not been explicitly considered in existing research. The usability designer in the case will typically be described as an "advocate of usability" (Boivie, Gulliksen et al. 2006), and she had gradually as a result of the internalization of inequality become less an advocate of usability and begun legitimatizing the status quo. Further personal skills such as having a great deal of stamina and a strong belief in what one is doing have been listed in (Boivie, Gulliksen et al. 2006) as prerequisites for usability designers in organizations where issues of usability are not prioritised. Our findings suggest that even a usability designer possessing these skills may be overshadowed by internalization of inequality. We suggest that the possibility of its occurring should not be overlooked, hence adding to the results in (Boivie, Gulliksen et al. 2006).

Thirdly, similarly on the part of the developers, previous research has observed that developers see users as similar to themselves (Bader and Nyce 1998), and that developers have an inadequate sense of usability of their own software (Høegh and Jensen 2008); and further that usability evaluations can debunk developers' erroneous assumptions of usability (Høegh and Jensen 2008). Our findings suggest that the effect of usability evaluation may not be realized. The developers in this study were well aware that there was a disparity in their perception of usability and that of their users. They also acknowledged that they did not have the necessary expertise to ensure usability, yet in spite of this they chose to justify the company's approach to usability work. Therefore it is necessary to be aware of the existence of such system justification tendencies in order to recognize their influences and to be able to discern when they are being used. With this finding we thus add to (Høegh and Jensen 2008) the awareness of potential system justification taking place. Along similar lines (Rajanen and Iivari 2007) argued for using the cost benefits of usability to justify changing the development process to include usability work; and our findings suggest that might be overtaken by a cover-up to preserve the status quo.

Fourthly, existing research development-centric companies are more receptive to the logic of usability and more inclined to move to a higher level of maturity if pushed (Nielsen 2006), and that usability designers need to be able to tailor their message and focus on selling usability (Iivari 2006). Our case and the analysis of it suggests that such an intervention or mere allusions to a company's problematic usability processes can be construed as criticisms of the organization and trigger system justifying tendencies hence with the preservation of status quo as a consequence. The implication of our study is however that we concur with (iivari 2006) that it is necessary to understand an organisation's culture fit usability processes to the

organizational culture. Based on our finding we further suggest that these organisational cultures are not necessarily easy to understand and that sometimes a deeper assessment of underlying justifications may be necessary.

Fifthly, the stereotypical attitudes used in our case company to describe usability designers and users are similar to what has been reported in previous research. Usability designers have been described as retrogressive by developers due to their focus on user studies prior to the start of the development (Norman 2006), usability designers being viewed as technically incompetent due to their lack of understanding of the development process (Kollmann, Sharp et al. 2009), that end users are being seen as peripherals (Shackel 2009), and that they should be kept at arm's length (Gulliksen 2007). Our findings show that such views were indeed also found in our case, but we suggest that these views are not inherent or intrinsic to software development, but may in stead be caused by a gradual rationalisation and justification for status quo and against change.

Sixthly, it is not uncommon to report from practice that software development companies do not find that integration of development processes and usability processes can be reconciled due to different cultures (Bygstad, Ghinea et al. 2008). Our findings corroborate this, but we suggest that we have added to a more detailed understanding of why that is – in some cases. This also correlates with the reports that the integration of usability work in development contexts is less a consequence of the method used but more dependent on how usability work is perceived within these contexts (Ferreira, Sharp et al. 2011; Wale-Kolade, Nielsen et al. 2013). Based on the developers' views in our case and their support of agile development processes we will question whether the agile culture have also triggered their system justification tendencies. There is nothing to suggest that in the case data.

Conclusion

In a case study of the integration of usability work into software development processes it has previously been reported that there is a gap between intention and reality . Our study confirms the presence of this gap in the case company, but it also points out that the gap can – as in our case – be based on fundamental justifications, not only presenting itself as a gap between what is intended and what is actually done. The case shows an example of a company where the perceptions of usability work were far from favourable. We have been seeking answers to the question of how they perceive usability work and we have provided evidence to show how they at one level praise usability, but in their practice they are not adhering to their own praise. In order to understand these contradictory attitudes we have used systems justification theory as a theoretical lens through which we have been able to show how their justifications have led to:

- Stereotyping of super users, end users and usability designers.
- The usability designer's internalisation of this stereotyped inequality.
- Rationalisation of status quo by the developers.

We have shown with quotations from interviews with the informants how the case organisation expresses this kind of justification. This is hugely incongruent. The findings can be used in at least two ways: (1) it highlights what a software manager, a usability designer, or a would-be integrator may well be up against; and (2) it points to how to potentially restore some congruence simply by demonstrating the severity of the implicit justifications leading to the explicit incongruence.

The limitation of the case study lies primarily in the use of system justification theory is that it does not tell when and how individuals' tendency to engage in system justification might be reversed (Gaucher and Jost 2011). We also see that the system justification theory does not address in any significant way a business perspective on the case company. Future research would benefit from studying the same phenomena in a business perspective and try to see to what extent and in which ways business rationality may lead the development team to refrain from going deeper into usability processes and practice. Another limitation of the presented case study is also that we have almost exclusively taken a process view; this has downplayed the issue of whether the developed system has a sufficient degree of usability and if that has been achieved implicitly without any explicit or systematic usability design processes. Hence, further research should be conducted to show how justification relates to perceived as well as measured usability.

References

Abrahão, S., N. Juristo, et al. (2010). "Interplay between usability and software development." Journal of Systems and Software **83**(11): 2015-2018.

Ardito, C., P. Buono, et al. (2013). "Investigating and promoting UX practice in industry: An experimental study." International Journal of Human-Computer Studies(0).

Bader, G. and J. M. Nyce (1998). "When only the self is real: theory and practice in the development community." SIGDOC Asterisk J. Comput. Doc. **22**(1): 5-10.

Benbasat, I., D. K. Goldstein, et al. (1987). "The case research strategy in studies of information systems." MIS Quarterly: 369-386.

Blasi, G. and J. T. Jost (2006). "System Justification Theory and Research: Implications for Law, Legal Advocacy, and Social Justice." California Law Review **94**(4): 1119-1168.

Blomkvist, S. (2005). Towards a Model for Bridging Agile Development and User-Centered Design. Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle. A. Seffah, J. Gulliksen and M. C. Desmarais, Springer Netherlands. **8:** 219-244.

Boivie, I., J. Gulliksen, et al. (2006). "The lonesome cowboy: A study of the usability designer role in systems development." Interacting with Computers **18**(4): 601-634.

Boivie, I., C. Åborg, et al. (2003). "Why usability gets lost or usability in in-house software development." Interacting with Computers **15**(4): 623-639.

Bygstad, B., G. Ghinea, et al. (2008). "Software development methods and usability: Perspectives from a survey in the software industry in Norway." Interacting with Computers **20**(3): 375-385.

Cajander, Å., J. Gulliksen, et al. (2006). Management perspectives on usability in a public authority: a case study. Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles, ACM.

Chamberlain, S., H. Sharp, et al. (2006). Towards a Framework for Integrating Agile Development and User-Centred Design. Extreme Programming and Agile Processes in Software Engineering. P. Abrahamsson, M. Marchesi and G. Succi, Springer Berlin / Heidelberg. **4044:** 143-153.

Chilana, P. K., J. O. Wobbrock, et al. (2010). Understanding usability practices in complex domains. Proceedings of the 28th international conference on Human factors in computing systems. Atlanta, Georgia, USA, ACM**:** 2337-2346.

Cockburn, A. and J. Highsmith (2001). "Agile software development, the people factor." Computer **34**(11): 131-133.

Dybå, T. and T. Dingsøyr (2008). "Empirical studies of agile software development: A systematic review." Information and Software Technology **50**(9-10): 833-859.

Ferreira, J., H. Sharp, et al. (2010). Values and Assumptions Shaping Agile Development and User Experience Design in Practice. Agile Processes in Software Engineering and Extreme Programming. A. Sillitti, A. Martin, X. Wang and E. Whitworth, Springer Berlin Heidelberg. **48:** 178-183.

Ferreira, J., H. Sharp, et al. (2011). "User experience design and agile development: managing cooperation through articulation work." Software: Practice and Experience **41**(9): 963-974.

Feygina, I., J. T. Jost, et al. (2010). "System Justification, the Denial of Global Warming, and the Possibility of "System-Sanctioned Change"." Personality and Social Psychology Bulletin **36**(3): 326-338.

Fitzgerald, B. (1998). "An empirical investigation into the adoption of systems development methodologies." Information & Management **34**(6): 317-328.

Fitzgerald, B., N. Russo, et al. (2002). Information systems development: Methods in action. London, McGraw-Hill Education.

Floyd, C. (1987). Outline of a paradigm shift in software engineering. Computers and Democracy: A Scandinavian Challenge. G. Bjerknes, P. Ehn & M. Kyng. Gower Publishing: 197.

Gaucher, D. and J. Jost (2011). Difficulties Awakening the Sense of Injustice and Overcoming Oppression: On the Soporific Effects of System Justification. Conflict, Interdependence, and Justice. P. T. Coleman, Springer New York. **11:** 227-246.

Gould, J.D. and C. Lewis (1985). "Designing for usability: key principles and what designers think." Communications of the ACM **28**(3): 300-311.

Gulliksen, J. (2007). How Do Developers Meet Users? – Attitudes and Processes in Software Development. Interactive Systems. Design, Specification, and Verification. G. Doherty and A. Blandford, Springer Berlin / Heidelberg. **4323:** 1-10.

Gulliksen, J., I. Boivie, et al. (2006). "Usability professionals--current practices and future development." Interacting with Computers **18**(4): 568-600.

Gulliksen, J., I. Boivie, et al. (2004). Making a difference: a survey of the usability profession in Sweden. Proceedings of the third Nordic conference on Human-computer interaction. Tampere, Finland, ACM**:** 207-215.

Göransson, B., J. Gulliksen, and I. Boivie (2003). "The usability design process -- integrating user-centered systems design in the software development process." Software Process: Improvement and Practice **8**(2): 111-131.

Haines, E. and J. Jost (2000). "Placating the Powerless: Effects of Legitimate and Illegitimate Explanation on Affect, Memory, and Stereotyping." Social Justice Research **13**(3): 219-236.

Hornbaek, K. and J. Stage (2006). "The Interplay Between Usability Evaluation and User Interaction Design." International Journal of Human-Computer Interaction **21**(2): 117-123.

Høegh, R. T. and J. J. Jensen (2008). "A case study of three software projects: can software developers anticipate the usability problems in their software?" Behaviour & Information Technology **27**(4): 307 - 312.

Iivari, N. (2006). "Representing the User' in software development--a cultural analysis of usability work in the product development context." Interacting with Computers **18**(4): 635-664.

Ji, Y. G. and M. H. Yun (2006). "Enhancing the Minority Discipline in the IT Industry: A Survey of Usability and User-Centered Design Practice." International Journal of Human-Computer Interaction **20**(2): 117-134.

Jost, J., Y. Kivetz, et al. (2005). "System-Justifying Functions of Complementary Regional and Ethnic Stereotypes: Cross-National Evidence." Social Justice Research **18**(3): 305-333.

Jost, J. T., M. R. Banaji, et al. (2004). "A Decade of System Justification Theory: Accumulated Evidence of Conscious and Unconscious Bolstering of the Status Quo." Political Psychology **25**(6): 881-919.

Jost, J. T. and O. Hunyady (2005). "Antecedents and Consequences of System-Justifying Ideologies." Current Directions in Psychological Science **14**(5): 260-265.

Kautz, K. (2011). "Investigating the design process: participatory design in agile software development." Information Technology & People **24**(3): 217-235.

Kay, A. C. and J. Friesen (2011). "On Social Stability and Social Change: Understanding When System Justification Does and Does Not Occur." Current Directions in Psychological Science **20**(6): 360-364.

Kay, A. C., D. Gaucher, et al. (2009). "Inequality, discrimination, and the power of the status quo: Direct evidence for a motivation to see the way things are as the way things should be." Journal of personality and social psychology, 97(3) **97**(3): 421-434.

Kay, A. C., M. C. Jimenez, et al. (2002). "Sour Grapes, Sweet Lemons, and the Anticipatory Rationalization of the Status Quo." Personality and Social Psychology Bulletin **28**(9): 1300-1312.

Kay, A. C. and J. T. Jost (2003). "Complementary Justice: Effects of "Poor but Happy" and "Poor but Honest" Stereotype Exemplars on System Justification and Implicit Activation of the Justice Motive." Journal of Personality and Social Psychology **85**(5): 823-837.

Kay, A. C., J. T. Jost, et al. (2005). "Victim Derogation and Victim Enhancement as Alternate Routes to System Justification." Psychological Science **16**(3): 240-246.

Kollmann, J., H. Sharp, et al. (2009). The Importance of Identity and Vision to User Experience Designers on Agile Projects. Agile Conference, 2009. AGILE '09.

Lárusdóttir, M. K., Å. Cajander, et al. (2012). The big picture of UX is missing in Scrum projects. . In proceedings of the 2nd international workshop on the interplay between user experience evaluation and software development, in conjunction with the 7th Nordic conference on human-computer interaction., Copenhagen, Denmark, University of Leicester, UK.

Lazar, J., J. H. Feng, et al. (2010). Research Methods in Human-Computer Interaction. United Kingdom, John Wiley & Sons.

Mao, J.-Y., K. Vredenburg, et al. (2005). "The state of user-centered design practice." Commun. ACM **48**(3): 105-109.

Miles, M. B. and A. M. Huberman (1994). Qualitative Data Analysis: An Expanded Sourcebook. United States of America, SAGE Publications.

Myers, M. D. and M. Newman (2007). "The qualitative interview in IS research: Examining the craft." Information and Organization **17**(1): 2-26.

Nandhakumar, J. and D. Avison, E. (1999). "The fiction of methodological development: a field study of information systems development." Information Technology & People **12**(2): 176-191.

Napier, J. L., A. N. Mandisodza, et al. (2006). "System Justification in Responding to the Poor and Displaced in the Aftermath of Hurricane Katrina." Analyses of Social Issues and Public Policy **6**(1): 57-73.

Nielsen, J. (2006). "Process Maturity." Retrieved 7 November, 2012, from http://www.useit.com/alertbox/maturity.html.

Norman, D. (2006). "Why doing user observations first is wrong." Interactions **13**(4): 50-ff.

Patton, M. Q. (2005). Qualitative Research. Encyclopedia of Statistics in Behavioral Science, John Wiley & Sons, Ltd.

Rajanen, M. and N. Iivari (2007). Usability cost-benefit analysis: how usability became a curse word?, Springer-Verlag.

Schwaber, K. and M. Beedle (2002). Agile software development with Scrum. Upper Saddle River: , Prentice Hall.

Seffah, A. and E. Metzker (2004). "The obstacles and myths of usability and software engineering." Commun. ACM **47**(12): 71-76.

Shackel, B. (2009). "Usability - Context, framework, definition, design and evaluation." Interacting with Computers **21**(5-6): 339-346.

Shepherd, S. and A. C. Kay (2012). "On the perpetuation of ignorance: system dependence, system justification, and the motivated avoidance of sociopolitical information." Journal of personality and social psychology, 97(3) **102**(2): 264-280.

Singh, R., L. Mathiassen, et al. (2010). "Sustainable Rural Telehealth Innovation: A Public Health Case Study." Health Services Research **45**(4): 985-1004.

Sinha, D. (2013). Frustrations in industrial work. Industrial Organizations and Health. F. Baker, P. J. M. McEwan and A. Sheldon, Taylor & Francis.

Stolterman, E. (1991). "How System Designers Think About Design and Methods:Some reflections based on an interview study." Scandinavian Journal of Information Systems **3**: 137-150.

Strode, D. E., S. L. Huff, et al. (2009). The Impact of Organizational Culture on Agile Method Use. System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on.

Thorisdottir, H., J. T. Jost, et al. (2009). On the Social and Pyschological Bases of Ideology and System Justification. Social and Psychological Bases of Ideology and System Justification J. T. Jost, A. C. Kay and H. Thorisdottir. New York, Oxford Universty Press**:** 3-23.

Venturi, G. and J. Troost (2004). Survey on the UCD integration in the industry. Proceedings of the third Nordic conference on Human-computer interaction. Tampere, Finland, ACM**:** 449-452.

Venturi, G., J. Troost, et al. (2006). "People, Organizations, and Processes: An Inquiry into the Adoption of User-Centered Design in Industry." International Journal of Human-Computer Interaction **21**(2): 219-238.

Vredenburg, K., J.-Y. Mao, et al. (2002). A survey of user-centered design practice. Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves. Minneapolis, Minnesota, USA, ACM**:** 471-478.

Vukelja, L., L. Müller, et al. (2007). Are Engineers Condemned to Design? A Survey on Software Engineering and UI Design in Switzerland. Human-Computer Interaction – INTERACT 2007. C. Baranauskas, P. Palanque, J. Abascal and S. D. Barbosa, Springer Berlin Heidelberg. **4663:** 555-568.

Wale-Kolade, A., P. Nielsen, et al. (2013). Usability Work in Agile Systems Development Practice: A Systematic Review. Building Sustainable Information Systems. H. Linger, J. Fisher, A. Barndenet al, Springer US**:** 569-582.

Walsham, G. (1995). "Interpretive Case Studies in Is Research: Nature and Method." European Journal of Information Systems **4**(2): pp.74-81.

Yin, R. K. (2013). Case Study Research: Design and Methods SAGE Publications Inc.

Zhang, P., N. Li, et al. (2009). "The intellectual advancement of human-computer interaction research: A critical assessment of the MIS literature (1990-2008)." AIS Transactions on Human-Computer Interaction **1**(3): 55-107.

Zhou, R., S. Huang, et al. (2008). A survey of user-centered design practice in China. Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on.