Abstract

Within agile software development there is a growing concern with how development organizations can integrate usability work into agile practices. The concern occurs as frustration experienced in practice by agile developers and usability designers, and it also occurs as a gap in the research literature. With this paper we report from a case study in a software company that is committed to agile development, to usability work, and to their integration. The theoretical starting point is an initial framework that has been elicited from the research literature; and the paper’s contribution is an extension and modification of the initial framework based on our case study results. The resulting framework points to three enablers (attitudes, compromises, skills), three tasks (upfront design, low-fi prototyping, iterative evaluating), and three alternative modes of collaboration/work organization (parallel, embedded, fully integrated) in agile usability practices. In addition, end-user involvement may vary. The paper contributes by extending existing frameworks on integration and thus providing better explanations for practitioners and researchers of integration of usability.

Keywords: Agile Development, Usability, Integration, Case Study.

1. Introduction

Usability work and agile software development both share focus on delivering value, are iterative in nature and engage in continuous testing [15, 17]. There are differences however in how they approach these issues, which leads to difficulties in integrating them in practice. In agile software development, value assessments are based on code quality and there is a single focus on speed of development [3, 2]. Conversely in usability work, value assessments are based on the quality and coherence of design and the approach to development is reflective and deliberate [2]. With evidence suggesting that that the integration of usability work into the agile development process leads to improved product quality and better user experience [17], how to ensure this integration has emerged as a major concern of research.

Research has been conducted on this culminating in a framework of practices that have been endorsed by researchers as critical for the integration of agile methods and usability work [27]. These practices cover issues such as what should be done, who should be involved, and how the development process should be coordinated. There have however been few examinations from practice or acknowledgement of the role situational considerations play in
ensuring this integration [27, 11]. A consequence of this is that there is a lack of clarity surrounding how this integration occurs in practice or what is critical for this integration [13]. In this paper we seek to contribute to a better understanding of the integration of usability work into agile development practices. By examining a software company we seek to answer the question: How is usability work integrated with agile practice and what characterizes this integration?

2. Related Research

Agile methods have over the last decade emerged as prominent approaches to software application development [7]. Their advent signaled a new era in software development. Unlike the rigid, engineering approaches that characterized the early years of systems development, these new methods were flexible, light weight and more amenable to changes in business requirements [8]. They prescribed an incremental and iterative approach to software application development which allowed for better management of requirements and also resulted in solutions that were more aligned with customers’ needs [16]. The two most popular agile methods used in practice are XP and Scrum and it has been observed that these two methods are often combined in practice [14], see Table 1. This is not to say that they are rigorously adhered to, as most organizations use tailored variants or in house methods whose application may involve one or more agile practices [7].

<table>
<thead>
<tr>
<th>Agile practice</th>
<th>Definition</th>
<th>Agile practice</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprints</td>
<td>Release new versions of the software in very short cycle</td>
<td>Simple design</td>
<td>The design should be as simple as possible</td>
</tr>
<tr>
<td>Sprint planning</td>
<td>Planning business priorities and technical estimates</td>
<td>Pair programming</td>
<td>Two developers on the same computer</td>
</tr>
<tr>
<td>Daily meetings</td>
<td>Short daily status meeting</td>
<td>Testing first</td>
<td>Write test code before writing function code</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>Reflect on strengths and weaknesses after each cycle</td>
<td>Continuous integration</td>
<td>Integrate and build the system when a task is completed.</td>
</tr>
<tr>
<td>40 hour week</td>
<td>Work time is generally limited to 40 hours per week</td>
<td>Collective ownership</td>
<td>Anyone can change any code anywhere in the system</td>
</tr>
<tr>
<td>On-site customer</td>
<td>Include an actual user on the team, available full time</td>
<td>Refactoring</td>
<td>Restructuring the software without removing functionality.</td>
</tr>
</tbody>
</table>

Table 2. Integrative practices [27]

For usability work this increased adoption of agile methods has come at a price. Agile methods present non-trivial constraints to the systematic performance of usability work [11]. Agile development favors just-in-time requirements over in-depth exploration of user requirements [3, 6], emphasizes rapid, iterative releases of software over sophisticated planning and design characterizing usability work [1, 9, 24], and advocates user involvement as customer involvement over the actual end users [3, 30]. This has led to sustained research interest into how these issues might be resolved with particular emphasis placed on how usability work and agile methods might be better integrated.

A recent literature review has condensed and highlighted the seven prescriptive practices, which researchers have advocated should be adhered to, if usability work is to be successfully integrated into agile software development [27], see Table 2.

<table>
<thead>
<tr>
<th>Integrative practices</th>
<th>Description</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uptfront design</td>
<td>Conduct activities such as contextual inquiry to identify usability requirements</td>
<td>Enables insight into use context and users’ perspective [3]</td>
</tr>
<tr>
<td>Low-fi prototypes</td>
<td>Translate identified user requirements into paper prototypes</td>
<td>Increases understanding of the user interface and usage [24]</td>
</tr>
<tr>
<td>Testing in between iterations with the application users</td>
<td>Test prototypes with real users to see if they meet usability and workflow requirements</td>
<td>Enables use focused refinements</td>
</tr>
<tr>
<td>Usability designers and developers work in parallel</td>
<td>Usability designers should work in a separate design track from the developers who similarly should have their own developer track</td>
<td>Enables synchronization of work flow</td>
</tr>
</tbody>
</table>
Usability designers should be involved in the project
Usability design team should be present to take on tasks relating to usability
Ensures usability concerns are always at the forefront

Usability designers should be fully integrated into the development team
Usability designers should participate in agile activities such as sprint planning, daily meetings
Facilitates easier sharing of findings from user research

End users should be involved in the project
Real users from the target system population should be involved in the project particularly during upfront analysis and testing
Ensures a reliable baseline for requirements predating to usability

The issue with these prescriptive practices is that integration is depicted as consistent and predictable [11]. It is an approach grounded on assumptions regarding how this integration occurs in practice and the context within which it occurs [27]. Further it fails to consider how processes are rarely followed methodically in practice [11].

There have been a few studies examining how development organizations integrate usability work into their agile development processes. Integration may be achieved in practice through articulation work, where usability designers and developers each step outside their boundaries to create opportunities for common action with one another [11] or may implicitly alignment work where they each assume responsibility for coordination and resolving integration tensions [4]. This requires mutual awareness of each other’s tasks, that there are expectations about acceptable behavior, that there is a negotiation of progress and that they engage with each other [13]. Organizational culture is also of some significance for integration [10, 12], and usability designers could be reduced to doing reactive work [21, 18]. Usability designers’ countermeasures can be to employ workshops to introduce usability work [5], be persuasive [5], and negotiate, communicate and be flexible about their beliefs [20].

These studies show that examining practice allows for a better understanding of the dependencies and mechanisms that make the integration of usability work and agile methods possible [13].

3. Case Study

Company XYZ is a Nordic IT service provider and independent partner of selected software vendors. It operates in four countries and employs 100 systems and software professionals. The organization has a focus on solution consulting, development and optimization. It has a strong foundation in enterprise and business intelligence systems. The company has two divisions of particular interest: the Portal & Solutions (P&S) division and the Business Intelligence (BI) division. The P&S division had considerable experience of more integrated usability work and agile development. The BI division on the other hand had recently completed a large project with a strong focus on usability.

3.1 Data Collection and Analysis

This is an interpretive case study based on qualitative data. Interpretative research allows us to see the agile practices and the integration practices as socially constructed and thus open to several interpretations by organizational actors but also to us as researchers [28,19,29]. With a case study we seek to explore the particulars in its organizational context with the potential of highlighting their practices and the circumstances that the case organization faces.

Data were collected from 7 informants, see Table 3. These informants were recommended by the contact person on the basis that they were well-informed about the issue being investigated and able to communicate about it. The interviews were semi-structured and followed the ideas and techniques laid out in [25] and conducted following an interview guide. The interview guide was designed based on the framework put forward in Section 2 summarized in Tables 1 and 2. These interviews which we conducted in English were audio recorded and later transcribed.
Table 3. Overview of interviews

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Division</th>
<th>Role</th>
<th>Interview length</th>
<th>Quotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anton &amp; Benny</td>
<td>BI</td>
<td>Domain expert</td>
<td>1 h 18 min</td>
<td>49</td>
</tr>
<tr>
<td>Cilla</td>
<td>P&amp;S</td>
<td>Usability designer 1 and front end developer</td>
<td>1 h 2 min</td>
<td>57</td>
</tr>
<tr>
<td>Dority</td>
<td>P&amp;S</td>
<td>Usability designer 2 and front end developer</td>
<td>1 h 6 min</td>
<td>34</td>
</tr>
<tr>
<td>Eric</td>
<td>P&amp;S</td>
<td>Product Manager</td>
<td>1 h</td>
<td>39</td>
</tr>
<tr>
<td>Frank</td>
<td>P&amp;S</td>
<td>Scrum master</td>
<td>1 h 10 min</td>
<td>55</td>
</tr>
<tr>
<td>George</td>
<td>P&amp;S</td>
<td>Usability designer 3 and front end developer</td>
<td>1 h</td>
<td>27</td>
</tr>
</tbody>
</table>

The interviews were analyzed using qualitative contents analysis [25,22]. The data were analyzed utilizing the framework in Tables 1 and 2. The transcripts were loaded into DeDoose, a cloud-based tool for qualitative data analysis. In the qualitative contents analysis the following procedure was applied:

1. One author coded all the empirical data based on the coding scheme (cf. Tables 1, 2, and 4).
2. The two other authors reviewed the coding.
3. Disagreements on selection of quotations, on particular coding, and on definitions of codes were gradually and iteratively resolved.
4. When agreement of the resulting coding was reached the analyses offered by DeDoose were used to create an overview of the empirical base. E.g., a coding frequency analysis was used to see the relative importance of codes in the empirical base. E.g., a co-occurrence analysis was used to see how the integration practices co-occurred with the agile practices.
5. To reduce the resulting codes into a smaller number of analytic units, the codes were examined for similarities. A pattern was observed and the codes were clustered together according to their observed similarities. Three categories emerged from this process.

Based on this contents analysis and the systematic application of the coding scheme the case presentation and the findings were elicited.

3.2 Overview of the Analysis

Table 4 provides an overview of the analysis. Based on the frameworks in Section 2 we see that there is much evidence of the integration of usability work into their development practices. The right hand side shows frequency of the occurrences of the seven integration claims, cf. Section 2, with the most prominent being ‘Designers partially involved’ (24) and ‘Testing between sprints’ (19). It also shows that the new code ‘Enablers’ emerged with a total of 44 occurrences - this emerging code will be explained in detail in the following Section. The left hand side of the Table shows frequency of occurrences of agile practices, cf. Section 2, with the most prominent being ‘Sprints’ (17) and ‘Planning game’ (17). A contents analysis shows that these occurrences contain evidence that these practices are actually being used in the case company to a large degree. To control for counter-evidence the code ‘Aberrations’ were applied to all quotations where the interviewees only talked about a practice without any evidence that it was in use. A cross-quotation analysis shows that these aberrations concern the less frequent practices, e.g., ‘Daily scrum’ (2).

The differences between the roles occupied by the interviewees have been analyzed. The main difference is that the usability designers refer more to integrative practices than the others and the software development managers refer more to agile practices. These are, however, not significant - with the only exception being the number of occurrences of
‘Enabler > Attitudes’ referred to by one usability designer in particular. Therefore, the
detailed analysis continues without making a strong distinction between the interviewees’
roles.

| Table 4. Overview of qualitative codes and frequency of evidence |
| --- | --- | --- | --- | --- |
| **Role** | **Domain expert** | **UX designer** | **Manager** | **Sum** |
| Agile practices (across all practices in [11], cf. Table 1) | 23 | 31 | 41 | 95 |
| Integration practices | | | | |
|Upfront design | 3 | 3 | 4 | 10 |
|Low-fi prototypes | 3 | 3 | 5 | 11 |
|Testing between sprints | 4 | 6 | 9 | 19 |
|Work in parallel | 1 | 2 | 1 | 4 |
|Designers partially involved | 8 | 9 | 7 | 24 |
|Designers fully integrated | 8 | 3 | 11 |
|End users involved | 1 | 3 | 5 | 9 |
|Enablers | 1 | 1 | 2 |
| - Attitude | 20 | 4 | 24 |
| - Compromises | 2 | 6 | 8 |
| - Skill sets | 9 | 1 | 10 |

3.3 Findings

Our study aimed to understand how usability work was being integrated in this company.
There are three main findings which emerged from the data analysis. These pertained to
enablers of integration, what tasks were being done and forms of collaboration.

3.3.1 Enablers

The usability designers refer much to enablers of integration (Table 4). Enablers are found
also in the other interviews across the different roles and divisions. Hence, the integration
does not only depend on the practices in the framework, it also depends on what can enable
projects and their developers, managers, and designers to further possible integration
practices. The following quotation shows an example of this:

“I think nowadays customers have realized the importance of usability, nowadays they really
want that new systems are not like the previous ones with bad user interface. They want that
[...] we work with usability from the start and they are very committed to my point of view to
that.” (Eric)

The quotation not only shows that usability is on the agenda in a positive way and that the
conditions with the clients enable this. It also shows more specifically that the attitudes of the
clients and of the product manager is part of enabling work being done on integrating
usability into the development project. Attitudes are central enablers, but sometimes the
necessary attitudes are absent as this quotation shows.

“Sometimes the developers’ attitudes is a problem, they don’t want to accept the fact that I do
know this better and sometimes they just have to learn, sort of the bad way, hard way to do
it.” (Cilla)

The analysis thus shows that it is relevant to include enablers in our understanding of how
usability work can be integrated with development. It further shows that there are three
aspects of enablers (Table 4). These are: (1) attitudes; (2) compromises; and (3) skill sets.
Attitudes occur more often than the two other, and without favorable attitudes many other
aspects of integration will face difficulties.

From this analysis we can now define ‘integration enablers’ as something that furthers or
assists in achieving integration of usability work into development practices. The claim
coming from the analysis of the case is that enablers are or can be important for integration.
We can also in more detail suggest that ‘attitudes’ are the perceptions held by involved actors (e.g., clients, users, designers, developers, and managers) on how necessary usability work is for the quality of the final product. From the literature we know that these actors may have different attitudes, and from this case we see that positive attitudes become important enablers. Along the same line we suggest that ‘compromise’ is an enabler that similarly influences the behavior of actors with the differences being that a compromise is a resolved disagreement on the relevance and usefulness of usability work. A different enabler is ‘skill set’ which we based on the case come to define as answers to which skills are needed to perform usability design and evaluation and in particular how to bridge the gaps between development and usability work.

3.3.2 Integrated Usability Practices

The data analysis shows three types of integrated usability practices were referred to: upfront design (10 occurrences), low-fi prototyping (11 occurrences) and testing between sprints (19 mentions; see Table 4). These practices were found in the case, but they varied greatly between projects and the two divisions.

In the P&S division upfront design covered methods to “get end-users on the loop” (Eric). This could involve, e.g., ‘wire framing with end users during the conceptualization phase.” Sometimes, the wireframes were also initiated by other project stakeholders than usability designers. For example, the customer could supply “wireframes models and sort of visual design” designed by another subcontractor. Dority mentioned that the graphic designers behind such designs often failed to think about “the interactive design” and “sort of drew beautiful layouts,” but did not think much about the consequences to functionality. Whenever this occurred, upfront design tasks would be extended to include an evaluation of the design concept, discussions on how the system-user interaction should occur, focusing on the end users’ point of view, and ensuring that both the implementation and the service design match to the user needs. Cilla also talked about how they also sometimes engaged in user research by carrying out user surveys and having “tight workshop(s)” where they went through ideas with the customer. This was not a standard practice as sometimes “the timetable would be so hectic that the customer would say, forget that we already know what we want and [there is] no need for that”.

As a contrast to upfront design, usability work in the BI project began with low-fi prototypes after the first initial version of the layout was delivered by the technical project team. The usability designers could not design a user interface for business intelligence applications from scratch and no upfront design was thus conducted. The reason for this was hinged on the application type and the expectations of the clients.

In P&S, the prototyping phase follows the upfront design phase. The team uses interactive wireframes as low-fi prototypes. Dority argued that these wireframes were more productive as if they were to “just describe the screens and then create sort of word documents where you describe all these things … it is easier to understand the overall whole thing.”

Testing and usability evaluation iteratively between sprints was mentioned several times in P&S, but it was much less frequently implemented in practice. Usability tests were usually done during the prototyping phase on the interactive prototypes, but little or no testing occurred during the development phase. Users were seldom involved in these tests and the usability designers typically carried out heuristic evaluations by themselves. The observed usability issues were fixed and the designs passed on to the developers. Frank noted that this was because most of the time their clients did not “have the budget for that (testing) all the time.” On rare occasions, customers “would use some other company that has expertise or specialized for usability testing” (Frank) upon project completion.

In the BI project, the clients were focused on functional tests on the application and preferred no usability testing between iterations. Usability testing was carried out at the end of the project. During these tests, the developers were placed in a separate room where they were able to observe the usability designers performing these tests. Many usability problems were
noted and the magnitude of these problems led one of the domain experts to make the following conclusion:

“So, the lesson here is that usability testing should be done; if there is only like resources to do it once or if it is too expensive, then it should be done in the middle of the project not at the end but the best case would probably be to do it iteratively a couple of times, maybe 5 times with the users to get the best result.”

3.3.3 Collaboration

The analysis shows that there are three forms of collaboration employed in the case company: working in parallel (4 occurrences), designers partially involved (24 occurrences), and designers fully integrated (11 occurrences). These three forms follow the framework at an overall level, cf. Section 2, but the case shows these in more detail that all three forms may be useful in the same company yet under differing circumstances.

In the P&S division, usability designers are partially involved, but often also fully integrated and no project is ever carried out without usability designers. The designers participate in the initial planning of the project where they show clients the return on investment they stand to gain by investing in usability work. The company has come to “understand the importance [of having] usability guys working all the way from the first meeting to the end of the live of the system” (Eric).

The usability designers are also front end developers in P&S. George reiterates on whether the relative importance of usability work is a consequence of their development tasks:

“If I just wouldn’t do any coding, but just working as a usability designer, it doesn’t change that I am working with same people and we have freedom to like share the tasks when we form some groups ... and if it is the case that I am only doing the usability design, I am still sure my voice would be heard, we would have meetings and I would say maybe you should consider this and that.”

In several P&S projects the usability designers were fully integrated into the team and participated in the division’s daily scrum meetings and planning meetings. Frank mentioned how during planning meetings that they “discuss every task and issue open[ly] together and what [they] should do” and made “sure that the coders [understood] what they need to do before the usability guy can do his or her stuff.” This process also applied the other way around as it was important that the developers and usability designers came to agreement on what was feasible in terms of design and of development.

The passing of designs between the usability designers and developers is organized to be executed in parallel. The usability designer ahead of a sprint specifies the functionalities from the wireframes that have been created. This gives specific details for the developers to create the view in the sprint. The aim is “to keep the design ahead of the development phase but not so that all the design is done before you start the development phase. The design is sort of done” (Cilla).

Conversely for the BI project, the involvement of the usability designers was not as extensive as in the P&S division. The BI division was involved in the project for about 100 or 150 days. The client on the other hand had hired the usability designers to participate in the project for only 10 days. Also the BI team and the usability designers never met in person. This arrangement meant that the usability designers worked in parallel with the BI team and were thus not partially involved or fully integrated:

“The whole time they were available the whole project was pretty much ten days, so it was maybe three days and couple of calls, they would call me for example and ask if it is possible to do this or some kind of fonts or stuff like that and we just went through it on the phone and that was pretty much it”. (Domain expert)
There was a downside to this arrangement as despite the usability designers’ familiarity with the platform, they ended up “coming up with so much stuff” and their initial designs were “over the top”. The lean forms of communication between the BI team and the usability designers ended up being a critical element of their collaboration. As the usability designers passed on their design guidelines and designs to the BI team, the BI team would on their own part evaluate the feasibility of these recommendations. If issues were noted, they would communicate these to the usability designers by phone and email.

“But when the graphic designer doesn’t know if it is possible he can do something like this and if we can implement this then we will but if not then we won’t”. (Domain expert)

This somewhat parallel collaboration between the BI team and the usability designers turned out to be effective as the domain experts noted that, “we had good discussions with them and now we are going to use them more and more”.

4. Discussion

Our findings suggest altogether three contributions to the previous literature on integrated usability practices in agile development which will be discussed below:

1. Identification of the three enablers (attitudes, compromises, and skills) as necessary for practice integration
2. Three concrete usability design tasks varyingly integrated to agile development: upfront design, low-fi prototyping and iterative evaluation
3. Three modes of collaboration between usability designers and developers (parallel, embedded and fully integrated) and end-user involvement varyingly observed in relation to the integrated usability design tasks

The previous literature reviews on integrating usability practices with agile development have been focused on highlighting the process model underlining existing approaches to integration [26] and on the motivations, warrants, theoretical backing and reported challenges (i.e., rebuttals) of integration [27]. The identification of the three enablers for integration provides a new category of phenomena, which warrants future research and managerial attention. Whereas previous case studies have identified challenges for integrated usability work in agile development milieus [18]. Especially, the categories ‘attitude’ and ‘compromise’ which we observed in the data seem to highlight importance of working with organizational culture. Recognizing and facilitating necessary changes in organizational culture for integrated usability and agile design are needed in the management of groups involving agile developers and usability designers. As well, more detailed, perhaps even professionally verified and standardized definitions of skill sets for the “agile usability designer” of the future should be developed and verified. More explicitly defined skill sets would enhance education and competence development of the future professionals, both developers and usability experts, to become familiar and to adopt more integrated practices.

Three general-level, previously identified practices of integrated usability development: upfront design, low-fi prototyping and iterative usability evaluation [27], were varyingly present in our target organization. And that’s about it: we found no traces of other concrete usability tasks in our case! Low-fi prototyping seemed to be a practice which fitted to all projects in our case, while upfront design was found less suitable in the business intelligence project. Whereas the most projects in our target organization seemed to skip iterative testing and evaluation between sprints and during development, the developers and usability experts seemed still to wish iterative usability testing to be increasingly used in the future. Our results also address that end-users were not always involved in the “agile usability” practice, but the iterative evaluation could take place plainly heuristically with “skilled usability experts”. Thus, iterative evaluation actually took sometimes place, but often without end users. In some occasions end users were involved only in the very beginning and in the end, and sometimes even less. Whereas our informants identified the need for and potential benefits of end-user involvement and more iterative testing, and would probably have re-produced the ideal “key
practices” as identified by [23] - feedback days, user workshops and micro-testing - in a general-level interview, our research shows that the actual practice may vary even within one organization and may skip end-user involvement for various reasons. More in-depth, perhaps action-oriented, research will be needed to reveal how end-user involvement and more iterative usability work during actual software development could really work more frequently in action, instead of plainly popping up as rhetorically recognized “ideals” in the professional field studies and practitioner interviews.

The case study enlightened us to re-categorize three of the previously observed “integrative usability practices” [27]: usability designers and developers working separately, usability designers as part of development projects, and usability designers as fully integrated to development - as alternative modes of collaboration between developers and usability designers. In our case we found evidence of all the three modes of organizing collaboration. In our case, the usability designers recognized the role of their tasks when integrated especially in relation to (and sometimes within) the daily scrum and sprint practices. In previous literature and cases [11], the prevailing mode of collaboration seems to be that the usability designers and developers represent separate work roles and work rather separately in parallel to the development, just handing over the usability specifications before and between sprints to the developers. While we saw that such parallel practice had been in use in some projects, the designers and developers had also chosen tighter modes of collaboration. Especially, the P & S division had recognized the benefits from more embedded and even integrated collaboration forms among usability designers and developers. Further research will be needed whether explication of collaboration modes between designers and developers is only needed in relation to daily scrum, sprints or when a developer is simultaneously usability professional in connection to pair programming, or whether usability work should also be more visible in other types of agile practices as well? Such research, however, requires in-depth fieldwork, perhaps even experimentation, for observing which practices would really work in action, how and why.

5. Conclusion

This article represented a case study on how usability work was integrated to agile development practices in a systems development organization, which had explicitly committed itself to these ideals. The data analysis highlighted the importance of three enablers, attitudes, compromises, and skills, as necessary prerequisites for reaching integrated and agile usability practices. However, despite its commitment on both agile development and usability, and while possessing already good experiences from such integrative practices as low-fi prototyping and upfront design, our target organization still poses a great variance in everyday practice with regard to actual implementation of iterative usability testing/evaluation and end-user involvement. As well, the organizing collaboration between developers and usability designers could vary greatly from project to project - while the overall ideal was recognized to be a more embedded, if not fully integrated, mode of collaboration. The study addresses a need further research on the enablers of integrated practice, such as on how to facilitate organizational culture on attitudes and compromising between sometimes deviating preferences of usability designers and developers, as well as on how to develop skill sets among the professionals towards the more “integrated” practitioners. More in-depth research is also needed on how iterative usability testing and evaluation practices together with increased end-user involvement could move from words to deeds.

References


