Open Source and Human Computer Interaction Philosophies in Open Source Projects – Incompatible or Co-Existent?

Mikko Rajanen
Dept. of Information Processing Science
University of Oulu
PO Box 3000, Oulu, Finland
mikko.rajanen@oulu.fi

Netta Iivari
Dept. of Information Processing Science
University of Oulu
PO Box 3000, Oulu, Finland
netta.iivari@oulu.fi

ABSTRACT
The importance of introducing usability activities into open source software (OSS) development has been acknowledged in the research literature; however, there is a lack of research examining the compatibility and actual co-existence of the philosophies of OSS and Human Computer Interaction (HCI), guiding usability research and practice, in OSS projects. This paper fills in this gap through an empirical examination of two OSS projects into which usability activities were introduced. The results show that in these cases the usability specialists embraced aspects of both philosophies; hence, these philosophies co-existed. However, the usability specialists either 'became them' or 'were close while kept the distance'. In both cases the usability specialists aligned their work with the OSS philosophy. However, through this alignment, in the other case they become very immersed in the OSS project, encountering a risk of becoming misaligned with the core HCI philosophy. Implications for research and practice are discussed.

Categories and Subject Descriptors
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Design, Human Factors.

Keywords
Open source software, Human Computer Interaction, Usability

1. INTRODUCTION
This paper examines the introduction of usability activities into open source software (OSS) development projects, and by doing so focuses especially on the issue of combining core Human Computer Interaction (HCI) and OSS philosophies in OSS development, HCI philosophy guiding usability related research and practice, OSS philosophy that of OSS development. Philosophy is in this paper understood as a "theory or attitude that acts as a guiding principle for behaviour" (http://oxforddictionaries.com/). The paper inquiries the nature of these two philosophies, guiding the behaviour of OSS developers and HCI/usability practitioners, as well as the potential points of friction between these two philosophies and the implications of their potential co-existence in OSS projects.

The HCI literature recommends that when introducing usability activities into software development, one should gain a thorough understanding of the context into which the activities are to be introduced in order to select the most suitable approach, as no 'one size fits all' [3, 22, 23]. Developers are the most important target group for usability activities and usability specialists should therefore make the developers perceive them as useful contributors and allies [3, 8, 17, 29, 40, 43]. Also in the OSS development context, usability specialists have been seen as needed, whereas their position has been reported of being quite challenging [2, 10, 33, 47]. Usability specialists tend to work in isolation in OSS projects, their work not having any impact on the OSS solution under development, usability specialists also having difficulties in showing their merits and in building reputation in OSS projects [2, 4, 5, 32]. It has been argued that usability activities should be tailored to fit the OSS development philosophy and culture [7, 9, 10, 33, 46, 49].

This paper is interested particularly in this 'fitting' between usability work and OSS development, the paper first examining the basic principles behind HCI and OSS development and then inquiring their potential co-existence in selected OSS development projects. The specific research question of this paper is: "How can the OSS philosophy and the HCI philosophy co-exist in OSS development projects?" So far, the existing research has reported on some benefits and challenges involved with introducing usability work into OSS development and argued for usability specialists to get to know and interact with the OSS projects they are trying to enter into [2, 4, 5, 7, 9, 10, 32, 33, 36, 37, 46, 49], however, there is a lack of empirical studies examining the co-existence of these two divergent philosophies, of OSS and HCI, and its implications in OSS development projects. It is obvious that the OSS and HCI philosophies are not entirely incompatible, as some usability activities have been identified in OSS development projects, while it is also clear that these two philosophies are not totally compatible either, as problems and challenges have been already reported in the

AcademicMindTrek'13, October 01-04 2013, Tampere, Finland.

literature and as on a more general level there are some clearly incompatible aspects in these philosophies. OSS philosophy places emphasis on technical development and gaining merit by contributing code [38, 39, 44] while the entire field of HCI has emerged to safeguard users who are neglected within technical development [12]. It has been warned that developers represent a particular kind of species, *homo logicus*, who nonetheless create solutions for the *homo sapiens* [11]. The field of HCI is needed to ensure that the homo logicus has some contact to homo sapiens while designing for the homo sapiens. In OSS development one usually gets merit and authority through showing competence in technical development, while experts in HCI do not usually possess that [1, 2, 4, 5, 32]. Therefore, there is an interesting research challenge regarding understanding how these two philosophies can co-exist in OSS development, this being especially important challenge from the HCI research point of view.

The paper is structured as follows. The next section presents related HCI and OSS research describing what the core philosophy within each case entails. The third section presents the research method utilized in this study as well as the procedures of data gathering and analysis. The fourth section presents the results of empirical analysis, the final section summarizing the results, discussing their implications and limitations and identifying paths for future work.

2. LITERATURE REVIEW

The following sections will discuss the HCI and OSS literatures describing the essence of the respective core HCI and OSS philosophies.

2.1 Core HCI Philosophy

This section reviews some basic HCI literature prescribing the work practice of usability specialists. The field of HCI gained its original form during the 1980s, during which the field focused on the generic aspects of human-computer interaction and on the 'look and feel' of software [20]. According to Bannon [6], the tradition of psychology, and specially the fields of human factors and ergonomics, which have for long considered the issue of how to develop technologies that are suitable for human beings, was very influential during these early days. As computer programmers and users became two separate groups of people with clearly distinct characteristics, skills and preferences and personal computers become widely used in people's everyday life, it became evident that HCI research and practice is needed. This field targeted its efforts on making computers easy to use for their users. The initial psychology orientation led to laboratory based experimental research. [6] For improving the ease of use, analytical models and design principles were developed. Also the first usability specialists entered the scene, working with user interface issues and carrying out laboratory based usability testing [24, 41].

Cooper and Bowers [12] have analyzed HCI literature as a discourse, showing how the field has legitimized its existence through the rhetoric on 'representing the user'. Actually they separate a first wave and a second wave HCI discourse, the first one legitimizing the existence of the field of HCI (including the expert group of usability specialists), utilizing even a 'political–war discourse' in places to convince that HCI is needed for fighting for the users that are neglected by computer science and systems design otherwise. However, a second wave HCI discourse emerged to criticize the first wave HCI discourse that was blamed of not producing usable results [12]. Criticism towards viewing users only as human factors, not as active human actors, was expressed and calls for more participatory design process expressed. Moreover, the interest turned to field studies and to the broader context of use and collaboration among users [6, 12, 41]. Therefore, the original focus on users and easy to use user interfaces was extended to cover also more participatory design process involving users and the broader acknowledgement of the tasks, goals, varying contexts and other users influencing use and usability that should be taken into account already during the design process.

From these premises, a number of usability methodologies (see e.g. [30, 34, 41]) have been developed, outlining the steps needed for developing and ensuring usability in the design process. Usually those all emphasize that first and foremost one needs to understand and specify who will be the users of the system under development. One needs to carefully consider the skills and characterizes of the future users, as well as their tasks, goals and the context of use. Thereafter, one needs to create design solutions deciphering how the users will behave in the future, when the system is in use, describing users' future tasks and work practices as well as human computer interaction solutions. Moreover, one needs to continuously evaluate the solutions to ensure that the solutions end up in being usable for the intended users. All this work involves user contact: in order to understand users and to evaluate the solutions from the users' point of view, users need to be involved in the development. Some even claim that users should be invited to the design process to figure out appropriate, usable design solutions together with designers (e.g. [41]); while not all usability methodologies share this view.

Usability specialists are the professionals responsible for usability in the development. Therefore, they are expected to gain understanding of the future users and gather their feedback to the made design decisions, and possibly even to invite users into joint design sessions. During these activities, users are placed into informative, consultative and possibly participative roles in the development, providing information and feedback as well as possibly taking part in the design process as equal participants with decision-making power regarding the design solution (cf. [13]). The same role repertoire actually applies to the work of usability specialists: they may be placed into informative and consultative roles, providing information and feedback to the development, while they usually wish to gain the participative role meaning that they act as equal design participants to developers, having decision-making power regarding the design solution, if not even gaining the position of a designer, being allowed to solely make important design decisions concerning the product quality [22, 11]. In addition to these roles, the essential task of the usability specialists is nonetheless to 'represent the user' in the development [12], i.e. to 'know the user' and to 'speak for the user' in the development [22]. In the philosophy of HCI the usability specialists gain merit from knowing the users and their needs, as well as developing and evaluating the user interface design solutions. The main characteristics of the HCI philosophy are presented in Table 1.

Table 1. The characteristics of the HCI philosophy

HCI Philosophy	Usability specialists are user representatives, They act in informative, consultative, participative and designer roles, sticking with the user focus, knowing the user, speaking for

	and fighting for the user in the development, Having decision-making power regarding important design solutions

2.2 Core OSS Philosophy

OSS refers to software whose source code is open for anyone to use, modify and distribute (see http://opensource.org/docs/osd). The OSS phenomenon is not new: actually the free sharing and modifying of software dates back to the emergence of software in general. Software communities that can be compared to modern free/open source software communities have existed for a long time before the terms "free software" or "open source software" were coined [25]. Software was not originally seen as a business asset like hardware was, and users of the software were generally capable of modifying the software themselves. Later on, the situation changed so that the value of software was recognized and proprietary software emerged. On the other hand, there also emerged strong reaction against this, arguing for free software and software sharing. This reaction can be traced back to the hacker culture of the 1960s and to the communities involved with UNIX operating system and C programming language development work. The Free Software Movement during the 1980s continues this trend as well as the Open Source Software Initiative of the 1990s. [19, 39]

OSS development can be characterized by its philosophical underpinnings, although the Free Software Movement was more ideologically oriented and the Open Source Software term can be considered as a reaction against it, instead highlighting issues relating to practical software development [19, 39]. Despite some ideological differences, the OSS and free software communities share many aspects, such as placing a high value on freedom of speech, regarding software as communal resources, and considering free information sharing as a right and an ideal [39, 45]. Also helping others so that they may solve new problems instead of readdressing old problems, and technical knowledge, skill and learning for its own sake are common values for both communities [39, 38, 44]. Altogether, from the texts characterizing OSS, the following basic values guiding OSS development have been identified: sharing, helping, technical knowledge, learning, voluntary collaboration and reputation [44]. Regarding the issues relating to the practical software development in the OSS context, Raymond [38] highlights e.g. the following issues: Important is that the development effort starts from a developer scratching his/her own itch, i.e. there is a personal motivation for the work. On the other hand, it is also important to attract other people to join in the effort as this enables rapid code improvement and efficient debugging. The 'release early, release often' phrase also highlights the importance of involving other developers and users in the continuous code improvement work, early and on continuous basis. The aim should not be to produce something finished during the first iteration but instead constant redesign and redevelopment work is believed to lead to good solutions. [38]

OSS development is usually organized as a loosely coupled community that is kept together by strong common values, the work being kept together by one or a few coordinators or core-developers [26]. In a typical OSS community, there is a lead core-developer or a small group of core-developers forming a core team that controls the overall architectural design and the course of the project [17, 31]. OSS community is often depicted with an onion model where different layers in onion represent the level of involvement within that particular OSS community. The core-developers or project leaders making decisions in the project form the hard core of the onion. In the next onion layer are the developers or code committers, who have direct read and write access to the project's source code and who support the core-developers. These developers have to ask permission of the core-developers to perform any major modification to the software. The contributors, who are external developers and users who send bug reports or minor fixes, form the next onion layer. These contributors can access the source code but they cannot upload their modifications directly to the project's source code repository. Their modifications are inspected by core-developers or developers, who upload the modifications to the code repository if they consider them beneficial and of good quality. The outmost onion layer consists of end-users, who do not participate actively in the development, but who use the software and may participate in the community. OSS communities are typically meritocratic and advancement through onion layers is considered as a reward and community recognition of a particular member's merits and abilities, especially relating to producing good quality code. [1] The main characteristics of the OSS philosophy are presented in Table 2.

Table 2. The characteristics of the OSS philosophy

OSS Philosophy	Software as communal resource, Voluntary collaboration, Loosely coupled community interaction, Different levels of involvement and status, Gaining merit and reputation through contributing to the community, especially high quality code, Scratching your own itch, Talk is cheap, show me the code

3. RESEARCH DESIGN

This research is part of a larger research program within which suitable methods and models for introducing usability work into OSS development have been developed and experimented with. The research program relies on design science approach, which is a type of scientific research that aims to develop new or improved ways to achieve human goals (c.f. [28, 21]). Design science research consists of two activities, building and evaluating. Building is a process for constructing an artifact for a specified purpose, and evaluating is a process of determining how well the constructed artifact performs in that specified purpose [28]. Building and evaluating information technology artifacts has design science intent [28]. In case the constructed artifact is totally new, the contribution of the research comes from the novelty of the artifact and persuasiveness of the claims that it is effective (c.f. [28]). In case the construct has already existed in some shape or form, the contribution of the research lies in the new form or shape of the construct being in some way better than the old one (c.f. [28]). Design science research involves a rigorous iterative and incremental process to design artifacts. In this case the artifacts have been methods and models for introducing usability into OSS development.

In this specific paper, however, the focus is on the core philosophies of HCI and OSS to find out how they can exist within a same instantiation. This focus is studied through a case study of two OSS development cases into which usability activities have been introduced (reported in [36, 37]). Case study

research is an empirical enquiry in which the focus is on a contemporary phenomenon in its real-life context when the boundaries between the phenomenon and its context are not clearly evident, and in which multiple sources of evidence are used [48]. In case study research, theoretical propositions are used as sensitizing devices to guide the collection and analysis of the data and typically multiple sources of data are preferable [48]. Here, the cases are two OSS development projects, into which two different university student projects have introduced usability activities under the guidance of HCI researchers. All students had multiple theoretical and practical HCI courses as a background. The student project members will be called usability specialists in this paper. The cases were selected for further analysis as the university student projects can be considered as relatively successful in both bases, revealing that the usability specialists succeeded in impacting the usability of the solutions in question. Now the focus is on how this happened.

The student project members acting as usability specialists communicated with the OSS communities and gathered research data, such as email, chat and forum messages between the usability specialists and the core-developers. They also stored all the reports of the usability activities and the redesigned user interface as research material. This research data was analyzed by the HCI researchers from the viewpoint of both HCI and OSS core philosophies; what kind of aspects relating to both of these philosophies could be found from the work of the usability specialists in each case.

4. EMPIRICAL RESULTS

The following sections will present the empirical results of each case, focusing especially on the findings relating to the reliance on the core HCI or OSS philosophies in each case.

4.1 Case 1: being close, while keeping the distance

Case 1 is an OSS community with 15 developers and relatively small user base of about 1000 users developing a game. The OSS community was relatively quiet during the usability specialist involvement but the community welcomed the usability specialists like all those who were willing to contribute to the community in some way or who were interested in the game in general. There was no previous usability specialist involvement with this community.

4.1.1 Being True to the Core HCI Philosophy

In this case the usability specialists conducted certain widely known and recommended usability activities in order to improve the usability of the OSS in question: they carried out both empirical user testing and expert usability inspections as well as produced redesign solutions that were meant to remove some of the identified usability problems. All these activities are recommended by the existing HCI literature [15, 27, 30, 34, 40, 42]. First the usability specialists carried out usability testing with four real non-technical users and a heuristic evaluation by combining several game heuristics [15, 16] and sent a report of usability findings and recommendations as well as an introductory document where the usability specialists introduced themselves and described their goal of improving the usability of this software. The usability team also sent the developers some redesigned user interface paper prototype mockups. Based on the feedback from the core-developers, the usability specialists decided to do a cognitive walkthrough to the level editor part of the game and make concept designs of the possible new user interface redesign fixing the usability problems found in the evaluation. The usability specialists wrote a report where the results also of this usability evaluation and the associated recommendations of the usability specialists were presented. All in all, one can conclude that the usability specialists acted in consultative and participative roles (cf. [22]) in the development, providing feedback as well as producing design solutions.

The usability specialists succeeded in having impact on the OSS under development. All the changes suggested by the usability specialists were made by the developers. This case also showed that the usability activities performed by the usability specialists were a successful wake-up call for the developers (cf. [43]) and interest in usability activities continued long after the usability team had finished its work. All in all, the feedback from the usability specialists was welcomed and it was taken seriously.

Interestingly, some of the developers expressed a view that the usability specialists should keep a certain distance from the core-developers and community in order to keep an objective view and focus on the users. In this case the usability specialists collaborated closely with the core-developers and community, but also kept a reasonable distance and remained as 'user representatives' speaking on behalf of the ignored users (cf. [12]).

4.1.2 Being True to the Core OSS Philosophy

Within the case, the usability specialists quite extensively tried to understand the context into which they were entering, as suggested by the extant literature on the matter [7, 9, 10, 33, 46, 49]. In addition, they quite extensively interacted with the OSS community in question in order to be able to show their merits and build reputation that are considered as crucial in OSS projects [2, 4, 5, 32]. At first, the contact with the developers consisted mostly of exchanging emails with the leading core developer of the project. As the case went on, the usability specialists sent documents and reports to the leading core developer as well as chatted on the project's Internet Relay Chat (IRC) channels with the whole community about usability and the potential benefits of the better usability as well as of the risks of poor usability, as recommended to be done when introducing usability into development (c.f. [35]). Although the importance of the leading core developer having a personal interest and approval for the usability activities in this case was evident, other members of the core development team clearly did not want to be left out and wanted their voices to be heard as well. In this case the OSS community was clearly interested in users and in improving the usability of their software, but a lack of knowledge also apparently existed regarding usability and its potential benefits. The assumption, altogether, was that the developers would be more willing to consider the input from the usability specialists if they had already started to become part of the community and had an established identity within the community.

The developers expressed that they would readily welcome any further usability help and especially hoped that they could receive this usability help by having a usability specialist as a close-knit part of their team and the game development, as a member of the development team. This is in line with the core OSS philosophy, but it presents a challenge to core HCI philosophy that highlights the importance of the usability specialists to act as boundary spanners between end users and developers. By being too closely involved with technical development the usability specialists may

be in risk of not being able to fulfill their original task of 'representing the user' in the development – in case their time and effort actually becomes invested into game development. On the other hand, also the HCI research literature recommends allying with the developers to ensure that usability work indeed impacts the solution under development. This involves usability specialists actively collaborating with developers, but without losing their original user focus.

4.2 Case 2: becoming them

Case 2 is an OSS community with twenty active core developers with commit rights and several other contributors and very active community including a forum with over 1000 active users and around 50000 posts. This OSS community had been actively developing a game since 2006. There was no previous usability specialist involvement with this community. The OSS community had a generally open and friendly culture welcoming all those who were willing to do something for the project or who were interested in the game in general and who did show skill and prowess in the game in particular.

4.2.1 Being True to the Core HCI Philosophy

Also in this case the usability specialists relied on standard usability activities such as different kinds of usability evaluations and redesign of the solution to remove some of the usability problems identified. In this case the usability specialists carried out heuristic evaluation and empirical usability testing. They also submitted a usability report based on their evaluations and proposed redesigned user interface mockups. The usability tests had real non-technical potential end users as test participants. The usability activities were extensive and altogether 156 potential usability problems were found through heuristic evaluation or usability testing with different types of test participants representing different kinds of potential or existing user groups.

Also in this case the usability specialists succeeded in having impact on the OSS under development. The usability report was referenced directly in commit messages four times and one of these commit messages asked for input from usability specialists after making changes based on the recommendations from them. The usability activities were continued over the time of two years with changing usability specialists bringing fresh outlook and focus to the evaluations. The developers thanked the good quality of the usability reports and even praised the work of the usability specialists as being of professional quality.

4.2.2 Being True to the Core OSS Philosophy

Also in this case the usability specialists quite extensively tried to understand the context into which they were entering as well as actively interacted with the OSS community, also in order to be able to show their merits and build reputation. Moreover, in this case the usability specialists contributed few code patches to the game and some were accepted by the core-developers and added to the official code repository. The usability specialists worked very closely with the core-developers. One of the usability specialists even managed to achieve a developer status during the usability activities due to his contributions in game development, recognized skills as a player and community activities. This can be seen as a mark of a very successful strategy for usability specialists to get recognition and merit within an OSS community. This follows the core OSS philosophy of gaining merit within an OSS community through development skills and skills that the community values, but it blurs the distinction between independent usability specialists siding with the non-technical end users as emphasized within the HCI philosophy and contributing developers valued in core OSS philosophy. From the HCI point of view this incident can be seen as a bit problematic as this particular usability specialist spent more and more time and effort on coding and game mechanics development and less and less time and effort on users and usability. However, as in this case there were multiple usability specialists working as a team, this was not such as problem as the rest of the usability specialists maintained their focus on usability and non-technical end-users.

5. CONCLUDING DISCUSSION

This section summarizes the main results, discusses their implications and limitations and identifies paths for future work.

5.1 Summary of the Results

This paper empirically examined the potential of co-existence of two divergent philosophies of OSS and HCI in OSS development projects. The research question of this paper was: "How can the OSS philosophy and the HCI philosophy co-exist in OSS development projects?" The empirical results are summarized in Table 3.

Table 3. The Co-Existence of HCI and OSS Philosophies in the Case OSS Development Projects

	Case 1	**Case 2**
HCI Philosophy	Usability specialists as user representatives, in consultative and participative roles, sticking with the user focus	Usability specialists as user representatives, in consultative, participative and designer roles
OSS Philosophy	Community interaction, gaining merit and reputation, representing end users, acting as bug reporters and bug fixers, interacting with core developers	Community interaction, gaining merit and reputation, representing end users, acting as bug reporters, bug fixers as well as committers, interacting with core developers, adopting developer focus and developer and gamer roles

The results show that while introducing usability activities into OSS development projects, the usability specialists embraced aspects in line with each philosophy, of that of OSS and HCI, in each case project. Hence, the results imply that these philosophies o-existed in these cases. The usability specialists, in line with the basic HCI philosophy, represented the users in the development [12, 22, 23] and took part in the design process in consultative and participative roles [13, 22], i.e. as evaluators and redesigners of the solutions. In OSS terminology they represented the end users and acted as bug reporters and bug fixers [1], communicating with the core developers. Moreover, in the second case the usability specialists adopted even designer and developer roles, meaning that they gained authority to make by themselves important design decisions concerning the OSS and even implemented those design solutions, i.e. acted not only as bug

fixers and reporters but as committers [1]. In both cases the usability specialists put extra effort on aligning their work with the OSS development philosophy: they interacted with the OSS communities to gain understanding of the project in question and to gain merit and reputation in the eyes of the community members. However, through this alignment work, they encountered risks of becoming misaligned with the core HCI philosophy that emphasizes continuous user focus and user representation work (e.g. [12, 22, 23]. In the second case a usability specialist became so immersed with the community in question that he contributed code, gained committer rights and through these adopted clearly a developer focus instead that of the user that nonetheless should had been his focus from the beginning until the end. In the first case the OSS developers also wished that the usability specialists stuck with that focus i.e. provided objective data from the users' and usability point of views. However, in the second case the OSS developers welcomed one usability specialist as gamer and developer into the community but might have lost some HCI contribution along the way as the usability specialist 'became them' rather than 'was close while kept the distance'. Luckily, there were other usability specialists in this case taking care of user focus at that time, but this is not necessarily the case every time usability specialists enter OSS communities to contribute.

5.2 Implications

As mentioned, the HCI literature has recommended that when introducing usability activities into software development, one should gain a thorough understanding of the context into which the activities are to be introduced in order to select the most suitable approach, as no 'one size fits all' and that developers are the most important target group for usability activities and usability specialists should therefore make the developers perceive them as useful contributors and allies [3, 8, 17, 22, 23, 29, 30, 34, 40, 43]. The same has been suggested also in the OSS development context, where in addition to understanding and aligning with the context, usability specialists should also be able to show their merits and gain reputation [3, 4, 5, 7, 9, 10, 32, 33, 36, 37, 46, 49]. This study contributes to this line of research. It gives support to the earlier findings along with indicating how this aligning could be done as well as warning of some risks involved.

The focus on meritocracy through development and participation causes problems for usability specialists in OSS development projects [2, 4, 5, 32]. This is because usability reports, recommendations and design mock-ups might not necessarily be recognized within an OSS community as contributions and therefore the usability specialists might have to gain merit within the OSS community by non-usability activities in order for their usability activities to have an impact. We emphasize the importance of community – especially core-developer – interaction so that the usability specialists become knowledgeable of the project in question as well as recognized by the community as valuable contributors. If the usability specialists are capable of contributing code, they very likely become valued participants. However, we do not consider this as a necessary condition for usability specialists to gain reputation and merit in OSS projects. In any case they should be able to enter the community and deliver their usability contribution in a way that becomes valued by the OSS community. Another issue to be taken into account by usability specialists entering OSS communities is that the OSS core philosophy regards software as a communal resource, which might mean that the usability specialists might not have to convince just one or a couple of core-developers about the importance of usability activities, but having to convince the entire OSS community, as even a vocal minority within the community may stop or reverse the usability improvements.

On the other hand, we identified a risk that the usability specialists become too immersed in the OSS project – through adopting the developer position and attitude. Although it is recommended that the usability specialists should work closely with the OSS developers so that the usability activities have an impact, on the other hand, they should also keep their distance and not to get too involved with the development and lose their focus on non-technical users. There might be a risk that the deeper the usability specialists dig into the onion of the OSS community, the further they distance themselves from the real non-technical users, who are typically not participating in the OSS community but whom the usability specialists should always remember to 'represent' and to 'speak on behalf'. This highlights an interesting paradox between HCI and OSS core philosophies: the usability specialists should aim at gaining access to and merit in the eyes of the OSS developers so that their usability activities have an impact, but by doing so the usability specialists may have to sacrifice some aspects of their HCI philosophies relating to non-development focus and user-centeredness, because OSS communities do not generally consider usability deliverables as a way to gain merit within the OSS community.

5.3 Limitations

As a limitation of these case studies, the usability specialists were not involved with the OSS projects from the very beginning of the projects. However, this is usually the case with usability specialist involvement in OSS projects as the starting OSS projects are usually not visible or readily accessible to non-technical outsiders until one or two developer hobby has evolved into a community-supported development project. Also, the results are based on two very specific OSS project cases that both happened to be developing a game and it is unclear if the fact that these OSS projects were developing games affected the usability activities and usability specialist involvement in some way.

5.4 Paths for Future Work'

There is still a need for more research on the appropriate and culturally compatible ways of combining HCI and OSS core philosophies and practices. Also, we maintain that still more research needs to be done related to how to introduce usability activities into OSS development in general, and how to address different kinds of OSS projects (e.g. large communities, small communities, different product domains, with or without company involvement, with different leadership styles and hierarchical structures, etc.) Also, after introduction into the OSS development context, usability also needs to be institutionalized (c.f. [43]) and this needs still future research in the OSS context.

6. REFERENCES

[1] Aberdour, M., "Achieving Quality in Open Source Software", *IEEE Software*, 2007, 24:1, pp. 58-64.

[2] Andreasen, M., H. Nielsen, S. Schrøder, and J. Stage, "Usability in Open Source Software Development: Opinions and Practice," *Information Technology and Control*, 2006, 25:3A, pp. 303-312.

[3] Aucella, A. "Ensuring Success with Usability Engineering," *Interactions,* 1997, May + June, pp. 19-22.

[4] Bach, P., and J. Carroll, "FLOSS UX Design: An Analysis of User Experience Design in Firefox and OpenOffice.org," *In Proc. OSS 2009,* 2009, pp. 237-250.

[5] Bach, P., R. DeLine, and J. Carroll, "Designers Wanted: Participation and the User Experience in Open Source Software Development," *In Proc. CHI 2009,* 2009, pp. 985-994.

[6] Bannon, L. (1991). From human factors to human actors: The role of psychology and human-computer interaction studies in system design. In J. Greenbaum, & M. Kyng (Eds.), *Design at Work. Cooperative Design of Computer Systems* (pp. 25-44), New Jersey: Lawrence Erlbaum Associates.

[7] Benson, C., M. Müller-Prove, and J. Mzourek, "Professional usability in open source projects: GNOME, OpenOffice.org, NetBeans," *Extended Abstracts of the CHI2004,* 2004, pp. 1083-1084.

[8] Bloomer, S., and R. Croft, "Pitching Usability to Your Organization," 1997, *Interactions,* Nov.+ Dec., pp. 18-26.

[9] Bødker, S., L. Nielsen, and R. Orngreen, "Enabling user-centered design processes in open source communities," In *Proc. Human Computer Interaction International, Part I: Usability and Internationalization.* N. Aykin (ed.), LNCS 4559, 2007, pp. 10-18.

[10] Cetin, G., D. Verzulli, and S. Frings, "An Analysis of Involvement of HCI Experts in Distributed Software Development: Practical Issues," In *Proc. Human Computer Interaction International: Online Communities and Social Computing*, D. Schuler (ed.), LNCS 4564, 2007, pp. 32-40.

[11] Cooper, A., and R. Reimann, *About face 2.0: the essentials of interaction design.* Indianapolis: Wiley, 2003.

[12] Cooper, C. & Bowers, J. (1995). Representing the users: Notes on the disciplinary rhetoric of human-computer interaction. In P. Thomas (Ed.), *The Social and Interactional Dimensions of Human-Computer Interfaces* (pp. 48-66). Cambridge: Cambridge University Press.

[13] Damodaran, L., "User involvement in the systems design process – a practical guide for users," *Behaviour & Information Technology*, 1996, 15:16, pp. 363-377.

[14] Desurvire, H., M. Caplan, and J.A. Toth, "Using Heuristics to Evaluate the Playability of Games," *In Proc. CHI 2004 extended abstracts on Human factors in computing systems,* 2004, pp. 1509-1512.

[15] Dumas, J., and J. Redish, *A practical guide to usability testing*, Norwood: Ablex Publishing Corporation. 1993.

[16] Federoff, M., "Heuristics and usability guidelines for the creation and evaluation of fun in video games," 2002. Available at: http://melissafederoff.com/thesis.html , last accessed 18 April 2010.

[17] Fellenz, C. B., "Introducing Usability into Smaller Organizations," *Interactions,* 1997, 4:5, pp. 29-33.

[18] Feller, J. & Fitzgerald, B. (2000). A Framework Analysis of the Open Source Development Paradigm. In Proc. of 21st International Conference on Information Systems, December 10-13, 2000, Brisbane, Australia, pp. 58-69.

[19] Fogel, K. 2005. Producing Open Source Software: How to Run a Successful Free Software Project. O'Reilly Media: Sebastopol.

[20] Grudin, J. (1991). Interactive Systems: Bridging the Gaps between Developers and Users. *IEEE Computer 24(4)*, 59-69.

[21] Hevner, A.R., S.T. March and J. Park, "Design research in information systems research," *MIS Quarterly*, 28:1, 2004, pp. 75-105.

[22] Iivari, N., "Understanding the Work of an HCI Practitioner," *In Proc. 4th Nordic Conference on Human Computer Interaction,* A. Morch, K. Morgan, T. Bratteig, G. Ghosh, D. Svanaes (Eds.), 2006b, October 14-18, pp. 185-194.

[23] Iivari, N., Culturally compatible usability work - An interpretive case study on the relationship between usability work and its cultural context in software product development organization, *Journal of Organizational and End User Computing*, Vol. 22, No. 3, 2010, pp.40-65

[24] Karat, J., "Evolving the scope of user-centered design," *Communications of the ACM*, 1997, 40:7, pp. 33-38.

[25] Levy, S. 1984. Hackers. Penguin Books. Middlesex UK.

[26] Ljungberg, J. Open Source Movements as a Model for Organizing. *European Journal of Information Systems 9*, 4 (2000) 208-216.

[27] Mack, R., and J. Nielsen, "Executive summary," In *Usability inspection methods*, Nielsen, J. and Mack, R. (Eds.) New York: John Wiley & Sons. 1994, pp. 1-23.

[28] March, ST., and G.F. Smith, "Design and natural science research on information technology," *Decision support systems*, 1995, 15:4, pp. 251-266

[29] Mayhew, D., "Strategic Development of Usability Engineering Function," *Interactions*, 1999a, 6:5, pp. 27-34.

[30] Mayhew, D., *The usability engineering lifecycle: a practitioner's handbook for user interface design*, San Francisco: Morgan Kaufmann Publishers. 1999b.

[31] Mockus, A., Fielding, R.T. and Herbsled, J. 2000. A Case Study of Open Source Software Development: The Apache Server. Proceedings of the International Conference on Sofware Engineering (ICSE), pp. 263-272.

[32] Moghaddam, R., Twidale, M. and Bongen, K. 2011. Open source interface politics: identity, acceptance, trust, and lobbying. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems.* ACM, New York, NY, USA, 1723-1728.

[33] Nichols, D., and M. Twidale, "The Usability of Open Source Software," *First Monday*, 2003, 8:1.

[34] Nielsen, J., *Usability engineering.* Boston: Academic Press, 1993.

[35] Rajanen, M., and N. Iivari, "Usability Cost-Benefit Analysis: How Usability Became a Curse Word?," In *Proceedings of INTERACT 2007: 11th IFIP TC13 International Conference on Human-Computer Interaction*, C. Baranauskas, P. Palanque, J. Abascal, S. Barbosa (Eds.): Lecture Notes in Computer Science 4663, 2007, pp. 511-524.

[36] Rajanen, M., Iivari, N. and Anttila K.: Introducing Usability Activities into Open Source Software Development Projects

– Searching for a Suitable Approach. *Journal of Information Technology Theory and Application 12*, 4 (2011), 5-26.

[37] Rajanen, M., Iivari, N. and Keskitalo, E. 2012. Introducing usability activities into open source software development projects: a participative approach. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design* (NordiCHI '12). ACM, New York, NY, USA, 683-692.

[38] Raymond, E., *The Cathedral & the Bazaar: Musing on Linux and Open Source by an Accidental Revolutionary*, O'Reilly & Assoc. 1999.

[39] Rolandsson, B. Bergquist, M. and Ljungberg, J. 2009. Open source programmer's strategies to cope with ideological tensions. In Proc. International Conference on Organizational Learning, Knowledge and Capabilities (OLKC), April 26-28 2009, Amsterdam, The Netherlands.

[40] Rosenbaum, S., J.A. Rohn, and J. Humburg, "A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys," In *Proc. CHI 2000*, Turner, T., Szwillus, G., Czerwinski, M., Paterno, F., Pemberton, S. (Eds.), ACM, New York, 2000, pp. 337-344.

[41] Rosson, M., J. Carroll, *Usability Engineering: Scenario-based Development of Human-Computer Interaction.* San Francisco: Morgan-Kaufman, 2002.

[42] Rubin, J., *Handbook of usability testing: how to plan, design and conduct effective test*. John Wiley & Sons, 1994.

[43] Schaffer, E., *Institutionalization of Usability: A Step-by-Step Guide.* Addison-Wesley, Boston, 2004.

[44] Steward, K. & Gosain, S. (2006). The Impact of Ideology on Effectiveness in Open Source Software Development Teams. MIS Quarterly 30(2), 291-314.

[45] Szczepanska, A.M., Bergquist, M. and Ljungberg, J. 2003. Challenging Codes in Open Source/Free Software Discourse. Proceedings of IRIS 26. 2003

[46] Terry, M., Kay, M. and Lafreniere, B.: Perceptions and Practices of Usability in the Free/Open Source Software (FOSS) Community. *In Proc. Conference on Human Factors in Computing Systems*, (2010), 999-1008.

[47] Twidale, M. and D. Nichols, "Exploring Usability Discussions in Open Source Development," In *Proc. 38th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2005.

[48] Yin, R.K. Case Study Research: Design and Methods. 1994. SAGE Publications.

[49] Zhao, L., and F. Deek, "Improving Open Source Software Usability," In *Proc. of the 11th Americas Conference on Information Systems*, 2005, pp. 923-928.