JITTA

JOURNAL OF INFORMATION TECHNOLOGY THEORY AND APPLICATION

A Publication of the Association for Information Systems

Introducing Usability Activities into Open Source Software Development Projects—Searching for a Suitable Approach

Mikko Rajanen
Department of Information Processing Science
University of Oulu
mikko.rajanen@oulu.fi

Netta Iivari
Department of Information Processing Science
University of Oulu
netta.iivari@oulu.fi

Kaisa Anttila
Department of Information Processing Science
University of Oulu
kaisaa@mail.student.oulu.fi

Abstract:

Usability is an important quality characteristic of software products and systems. Different approaches for introducing usability activities into SW development have not yet been evaluated with respect to open source software (OSS). This paper tests the introduction of usability activities through four empirical case studies in the OSS development context. Case studies were carried out using four OSS projects that were not commercially supported. Empirical analysis suggests that usability specialists should become members of the OSS community, but, at the same time, should keep an objective view. The usability activities had substantially more impact when specialists introduced and carried them out as fellow members of the OSS community than as usability consultants from outside the community. The importance of management commitment for usability activities is discussed in the OSS context. The challenge of adapting usability and OSS development philosophies together should be researched further.

Keywords: usability, open source software

Volume 12, Issue 4, pp. 5-26, December 2011

Peter Axel Nielsen acted as the Senior Editor for this paper.

JOURNAL OF INFORMATION TECHNOLOGY THEORY AND APPLICATION

Introducing Usability Activities into Open Source Software Development Projects—Searching for a Suitable Approach

INTRODUCTION

This paper examines four case studies in which usability activities are introduced into open source software (OSS) development. The specific research question examined in the paper is: How should usability activities be introduced into OSS development? OSS development has gained increasing interest in information systems (IS) research during recent years (see, e.g., Fitzgerald 2006, Niederman et al. 2006), and OSS development has been argued to be highly influential in the software landscape of the future (Fitzgerald 2006). Estimating the number of users and the market share of OSS solutions is difficult because OSS can usually be downloaded freely and from numerous mirror sites and peer-to-peer networks. One of the most well-known Web-based repositories is SourceForge, a leading resource for OSS development and distribution. It has about 2.7 million developers, over 260.000 OSS development projects, a total number of users in all projects estimated at more than 46 million, and more than two million downloads from project repositories every day (SourceForge.net). At least twenty other source code repositories and resources also exist for OSS development and distribution.

The size of an OSS development project can range from a single developer working alone to code and use the application to massive OSS development projects that span years and involve hundreds of developers, e.g., Linux and OpenOffice.org. The latest version of the OpenOffice.org office application suite has an estimated 15–20 percent market share. In 2009, OpenOffice.org announced that the latest version had recorded one hundred million downloads within a year from their main download site, and this did not include downloads from mirror sites and peer-to-peer networks (OpenOffice.org). The Firefox Web browser has been downloaded over one billion times and has a 23 percent worldwide usage share for Web browsers (Mozilla.com). The Apache Web server software serves 55 percent of all websites in the world and has reached the 100 millionth website milestone (Apache.org). Therefore, the OSS phenomenon clearly is highly influential in the current software landscape (cf. Fitzgerald 2006).

The term *open source software* refers to software for which source code is freely available for everyone to read and modify. The fundamental idea of OSS is to enable software to evolve outside of commercial company restrictions by exploiting the participation of technically oriented contributors and users in the outside community (cf. Raymond 1999, Ye and Kishida 2003). Traditionally, software development is carried out in closed settings as closed source software development, where only few people can access the code. In traditional software development, the end-users can see and use only the binary form of the software. In contrast, OSS development makes it possible for the end-users to adapt software to their personal needs and to fix defects (Raymond 1999). Currently, companies have also started to use OSS in their business. OSS applications and development tools have been in common use for a long time, but now the utilization of the actual source code is also becoming popular. Since the availability of free, ready-made components can substantially reduce development costs, companies have recently started to participate in OSS communities and even to launch and build new communities for their products (Dahlander and Magnusson 2005, Fitzgerald 2006, Iivari et al. 2008, Niederman el al. 2006). The revenue models of OSS have been changed from purely support selling and loss leading to more comprehensive marketing and sales management, servicing, and implementation (Rajala et al. 2001). In addition to utilizing OSS in their development work, recent studies (cf. Ågerfalk and Fitzgerald 2008, Fitzgerald and Ågerfalk 2005) also suggest that outsourcing the development work to the OSS community provides a significant opportunity for companies to headhunt top developers from OSS projects, which is also one potential benefit for companies that could be gained through entering the OSS scene.

CONTRIBUTION

The findings presented in this paper contribute to IS research by experimenting with different approaches to introduce usability activities into software development in the OSS development context. To date, this type of research has not been carried out, but is clearly needed. This paper evaluates existing HCI approaches for introducing usability activities into software development in the OSS development context. The evaluation uses four case studies and experiments with three ways of introducing usability activities into OSS development projects. The research process is iterative and incremental. A traditional way to introduce usability into development has been to employ a consultative approach. This study tests this approach in two cases and finds that the consultative approach has very little impact. Two further case studies, which use a participative role instead of a consultative role, suggest that usability specialists should become recognized members of the OSS community rather than parachuting in as consultants. This study is the first to examine management commitment in connection to usability activities in the OSS development context. Furthermore, this study has practical implications related to using student groups as usability specialists to introduce usability into OSS projects and related to advising these usability infiltrations as a whole. The study provides evidence that usability activities in the OSS development context have greater impact when the usability specialists become members of the OSS community and act as peers and fellow members within the community rather than as consultants.

Usability has been discussed particularly with respect to the field of Human Computer Interaction (HCI), which has also introduced a number of methods for improving usability (e.g., Mayhew 1999a, Nielsen 1993, Rosson and Carroll 2002). Usability is identified as one of the main software product and system quality attributes in the international standard ISO 9126. It refers to the capability of the product to be understood, learned, and used by users, as well as to appeal to users when used under specified conditions (ISO/IEC 9126-1 2001). Standard ISO 9241-11 provides another common definition for usability: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO 13047 1999). The importance of usability has also been acknowledged in IS research, wherein the constructs of usefulness ("the degree to which a person believes that using a particular system would enhance his or her job performance" [Davis 1989, p. 320]) and ease of use ("the degree to which a person believes that using a particular system would be free of effort" [Davis 1989, p. 320]) have been postulated as essential factors for successful technology adoption (Davis 1989) and diffusion (Rogers 1995).

Traditionally, technically skilled developers have developed OSS for their own use, but OSS solutions now have increasingly more users who lack deep technical knowledge. The OSS solutions have probably always been useful, since OSS developers have usually produced tools for their own use, but usability has not traditionally aroused great interest in connection to OSS, probably because the developers haven't encountered any usability problems with these tools they have created themselves. However, the present paper is motivated by the fact that an increasing number of OSS solutions now have user populations that no longer consist solely of developer-users: most of the users of several popular OSS solutions, such as Firefox, may not be capable of adapting the software to their needs or fixing or reporting defects (Giuri et al. 2004). Improving OSS usability and bringing usability activities into OSS development have not been researched in detail, but as more non-developer users start to use OSS, the importance of these factors will rise. Even though recent studies have described some usability activities that have already been carried out in some OSS projects (Andreasen et al. 2006, Bach and Carroll 2009, Back et al. 2009, Terry et al. 2010), the current status of usability activities in OSS projects and usability of OSS still tends to be quite poor (e.g., Cetin et al. 2007, Nichols and Twidale 2003, Nichols and Twidale 2006, Zhao and Deek 2005, Zhao and Deek 2006), especially in small and medium-sized OSS projects and in OSS projects without company involvement.

Usability is argued to provide an important competitive edge in maturing software markets (Grudin 1991, Nielsen 1993, Rosson and Carroll 2002). Users can benefit from better usability through higher productivity when the most frequent tasks take less time and users make fewer errors (Ehrlich and Rohn 1994, Mayhew and Mantei 1994). The development company can benefit from better usability through reduction in time and resources needed for product development due to reduced need for changes in later development phases (Ehrlich and Rohn 1994, Bevan 2000, Donahue 2001, Mayhew and Mantei 1994). The development company can also gain benefits from better usability when they can use improved usability as a competitive edge and thereby increase their sales (Ehrlich and Rohn 1994, Bevan 2000, Donahue 2001, Karat 1994, Mayhew and Mantei 1994). If the development company provides some kind of product support, better usability can greatly reduce the number of contacts made to customer support and thereby again reduce costs (Ehrlich and Rohn 1994, Bevan 2000, Mayhew and Mantei 1994).

Introduction of usability activities even into traditional software development is often reported as challenging (Ohnemus 1996, Iivari 2006, Rajanen and Iivari 2007). A number of reasons can be outlined in relation to this. For example, cost benefit considerations and resource constraints have been emphasized. Resources required for usability activities need to be well-planned and budgeted (Aucella 1997, Mayhew 1999a, Nielsen 1993), and usability activities should not increase development costs and time (Bloomer and Croft 1997, Nielsen 1993, Rajanen and Iivari 2007). On the other hand, the context into which the usability activities are to be introduced needs to be thoroughly understood in order to select a suitable strategy: a "one size fits all" approach is not recommended (Allen 1995, Aucella 1997, Hutchings and Knox 1995, Rajanen 2006, Iivari 2006, Schaffer 2004). The introduction of usability activities into OSS development presumably is challenging as well, but question marks remain attached to the relevant factors helping or hindering it. This paper contributes by experimenting with different approaches for introducing usability activities into software development in the OSS development context. This kind of research has not been carried out so far, but is clearly needed.

This paper is organized as follows. The relevant usability literature for this paper is presented in the next section. The research method used in this study is presented in the third section. The fourth section presents the empirical cases investigated in this study and the procedures for data gathering and analysis. The final section summarizes and discloses the observations of the paper, outlines the limitations of this study, and suggests areas for future research.

USABILITY LITERATURE

Usability Activities

This section describes the various approaches of ensuring the usability of an information system. A number of usability engineering (UE) and user-centered design (UCD) methods have been developed within HCI research for improving usability (e.g., Gulliksen et al. 2003, ISO 13407 1999, Mayhew 1999b, Nielsen 1993, Rosson and Carroll 2002). For example, usability engineering (e.g., Nielsen 1993, Mayhew 1999), scenario-based design (e.g., Rosson and Carroll 1992), and goal-based interaction design (e.g., Cooper 1999, Cooper and Reimann 2003) methods have been developed and contain phases such as requirements analysis, activity design, information design, interaction design, prototyping, usability evaluation, and documentation design (Rosson and Carroll 2002) or research, modeling, requirements definition, framework definition, and refinement (Cooper and Reimann 2003).

Even though many differences are apparent in these methods, they also clearly share certain characteristics. The first phase (or phases) of the methods typically relates to gaining an understanding of the intended users and of their needs, characteristics, work practices, or tasks to be supported by the technology, as well as current problems and the context of use in which the system or product will be used (e.g., Cooper and Reimann 2003, Mayhew 1999b, Nielsen 1993, Rosson and Carroll 2002). After gaining this understanding, the next step is a careful redesigning of the users' work practices or tasks, trying to eliminate the identified problems and to address the identified needs. This redesign of users' work practices or tasks should be done before any software design begins (e.g., Cooper and Reimann 2003, Mayhew 1999b, Nielsen 1993, Rosson and Carroll 2002). Thereafter, user interface design and development is carried out by producing different kinds of design solutions. This can be done in iterative stages, by which the user interface evolves from rough design concepts to a complete user interface. The rough design concepts can be produced with pen and paper to allow quick brainstorming and testing of the user interface, with walkthroughs in early phases of user interface design, to prevent design from becoming "railroaded," which means that the user interface design is stuck to the first user interface concept, and further design concepts of the user interface are just variations of this one user interface concept.

Finally, usability evaluation is also a crucial element in these methods. It should be started very early, in order that the results can truly affect the design and that changes do not become expensive, which would be the case if the problems were discovered in a finished or an almost finished system. In the early phase, the emphasis is on gathering qualitative feedback, whereas during the later phases of development, one should also measure whether the requirements have been met (ISO 13407 1999, Nielsen 1993, Rubin 1994). Usability evaluation methods have been separated into empirical methods, which necessitate involvement of real or representative users, and into usability inspection methods, which do not necessitate user involvement, but instead rely on inspectors' expertise and experience and on guidelines, standards, heuristics, etc. (e.g., heuristic evaluation, Nielsen 1993, Shneiderman 1987; cognitive walkthrough, Wharton et al. 1992). Usability inspections are recommended, since they can be used during a relatively early phase and many of them are quite fast and easy to use (Mack and Nielsen 1994, Nielsen 1993). Empirical usability evaluation, however, is of utmost importance: it is essential to have real or intended users evaluate the system. The most common and fundamental empirical usability evaluation method is usability testing, in which representative users are invited to a usability laboratory to carry out realistic tasks by using the system. Typically, the users are also asked to think aloud, so that the usability evaluators can determine what the users are trying to do and why they do what they do (Dumas and Redish 1993, Mack and Nielsen 1994, Rubin 1994).

To carry out these activities, methods will include suggestions regarding hiring of a group of specialists, who are labeled, e.g., as usability, UCD, UE, or interaction design specialists (e.g., Cooper and Reimann 2003, Iivari 2006, Mayhew 1999b, Nielsen 1993, Rosson and Carroll 2002). These persons will be referred to as *usability specialists* in this paper. These specialists are expected to carry out or to facilitate the gaining of the understanding of the users, their tasks, the context of use, the user task redesign, the user interface design, and the iterative usability evaluations.

Introducing Usability Activities During Software Development

In this section, we will review some advice already offered for the introduction of usability activities into software development. Especially during the early days of incorporating usability activities into development, usability specialists usually carried out empirical usability tests in usability laboratories (Karat 1997, Rosson and Carroll 2002). However, even currently, usability testing remains the most widely used, the most successful, and the most efficient method for improving usability in software development (Gunther et al. 2001, Rosenbaum et al. 2000, Vredenburg et al. 2002). In addition, usability inspection methods are widely utilized in industry (Gunther et al. 2001, Rosenbaum et al. 2000, Vredenburg et al. 2002). Ordinarily, these are not as labor intensive and equipment requiring, and, therefore, expensive, as empirical usability testing; consequently, they can be more easily and quickly adopted in development. However, they do not provide results for usability of the solution that are as valid as

those obtained with empirical usability testing, and, therefore, inspection methods should not be seen as replacing empirical testing, but instead as something that should be used in combination (Mack and Nielsen 1994, Nielsen 1993).

These methods position usability specialists into a consultative role (cf. Damodaran 1996, Iivari 2006b) in development: they act as commentators of predefined design solutions, providing feedback to the development (cf. Damodaran 1996, Iivari 2006b). However, more recently researchers have criticized that, in the consultative role, usability specialists are trying too late and too ineffectively to contribute to development; i.e., they are pointing out problems too late to be able to make major changes, while they should have contributed earlier and more directly to the design during the requirements-specification and design phases (Aucella 1997, Borgholm and Madsen 1999, Bødker and Buur 2002, Rosson and Carroll 2002). They should function in a participative role in development (cf. Damodaran 1996, Iivari 2006b), actively cooperating with developers during requirements construction and design, as well as having some decision-making power regarding the solution (Aucella 1997, Bloomer and Croft 1997, Fellenz 1997, Mayhew 1999a, Mayhew 1999b, Tudor 1998).

Indeed, for introducing usability into software development, the literature suggests that the developers are a highly important target group (e.g., Aucella 1997, Bloomer and Croft 1997, Fellenz 1997, Mayhew 1999a, Schaffer 2004), who should perceive usability specialists as teammates and allies (e.g., Fellenz 1997, Mayhew 1999a, Rosenbaum et al. 2000). However, management commitment and support is also an important criterion for success (e.g., Fellenz 1997, Schaffer 2004). An executive champion is needed to provide leadership, resources, and coordination. This executive should be educated about usability in order to have an understanding of usability, in addition to that of the development process, and should be able to see how usability fits into the development process. The executive champion should be in a position that can influence other people and should be able to sell and evangelize usability within the organization (Schaffer 2004). In addition, professional usability specialists are important (e.g., Aucella 1997, Fellenz 1997, Mayhew 1999a, Vredenburg et al. 2002). Organizations should first hire external usability consultants who already have the training, tools, and team needed for usability work. The hires should also have good change-management skills for establishing an internal usability team inside the organization later on (Schaffer 2004). Usability should be sold to development organization—by the executive champion, as well as by the usability specialists (e.g., Mayhew 1999a, Schaffer 2004). Typically, a wake-up call highlighting the need for usability improvement is required. This wake-up call might be a train wreck (the product proves to be impractical and unusable, this being the most expensive type of wake-up call), executive or new staff insight, usability testing results, or expert review results. The wake-up call, overall, should point out that usability really needs to be improved (Schaffer 2004).

With respect to the initial experimentation with the usability activities in projects, the literature emphasizes the importance of achieving quick but visible results—one should be able to show the benefits achieved (Bloomer and Croft 1997, Mayhew 1999a, Rosenbaum et al. 2000)—and warns that cost-benefit tradeoffs may play a decisive role in the adoption of usability activities, meaning that low-cost methods might be preferable (Vredenburg et al. 2002). It is of importance that usability activities do not increase the development costs and time (Bloomer and Croft 1997). Not surprisingly, studies have shown that management perceives orderliness, objectivity, and control as key success factors, and from that perspective, usability can be seen as unclear and difficult to incorporate in the business (Cajander et al. 2006). The usability cost-benefit literature encourages using the potential benefits of better usability and risks of poor usability to sell the usability activities in the eyes of the executives and the management and to further integrate usability activities in development with the help of this usability champion at the management level (cf. Ehrlich and Rohn 1994, Mayhew and Bias 1994, Mauro 2005). On the other hand, care should be taken when promoting usability cost-benefit considerations because it might actually raise the visibility of the costs of usability and make the potential benefits of usability not sufficiently convincing, thereby harming the introduction of usability activities into software development (Rajanen and Iivari 2007).

The literature goes even further, outlining how to institutionalize usability after the initial startup activities presented above. This involves training the personnel, even though usability people can also be recruited. Internal staff, in any case, should be trained with respect to usability: the developers need knowledge on a more general level, while people involved in user interface design need more extensive training and skills. Usability strategy, methods, tools, and infrastructure are also needed, as are design guidelines and standards. Showcase projects, in which usability activities are piloted for refining the activities as well as for showing their value, should be carefully selected. The usability team should act as evangelists, trainers, and mentors; they should maintain respect and momentum, and they should carry out the usability activities in projects (Schaffer 2004.) The present paper will focus only on the initial activities related to introducing usability to OSS projects, for which the activities related to institutionalization of usability are not yet that relevant, although they are expected to become so in future OSS projects.

Some studies also stress the importance of thorough understanding of the context into which the usability activities are to be introduced for selection of a suitable approach (Iivari 2006a). Organizational and political contexts should be acknowledged (Allen 1995, Hutchings and Knox 1995). In addition, organizational culture should be taken into account: one should understand the usability myths, beliefs, values, and attitudes within the organizations, since these may act as cultural obstacles to usability work (Bloomer and Croft 1997, Iivari 2006a, Mayhew 1999a). Overall, it should be emphasized that no "one size fits all" solution exists (Aucella 1997, Iivari 2006a), but instead a cultural match should be found between the corporate culture and usability work (Iivari 2006a, Schaffer 2004). Therefore, usability specialists should align their ways of working with the engineers (Mayhew 1999a).

Usability and OSS Development

The term *OSS development* refers to a community-based development model in which technically oriented individuals develop solutions based on their personal needs and then voluntarily reveal their solutions for use and further development by others (von Hippel and von Krogh 2003, Raymond 1999, Ye and Kishida 2003). The OSS community is often described using an onion model, in which the decision-making project leaders form the onion core. In the next onion layer are the committers, who have direct read-and-write access to the project's source code. These committers support the project leaders, and they need the permission of the project leaders to perform any major modifications. The next onion layer consists of contributors, who are external developers and users who send bug reports and minor fixes, or patches, during OSS development. These contributors can access and modify source code, but they cannot upload their modifications directly to the project's source code repository; rather, they have to get a committer or project leader to inspect their code and upload it to the code repository if the modification is considered beneficial. The outmost onion layer consists of end-users, who do not actively participate in the development, but who use the software. Typically, OSS projects are meritocratic, and advancement through member types and onion layers is considered to be a reward and a recognition of a particular member's merits and abilities (Aberdour 2007). However, in some OSS projects, the majority of users may not be able to adapt the software or fix or report defects, and they should be classified as a user group that "consumes" rather than "contributes" (Giuri et al. 2004). For these users, usability activities might be needed.

In the OSS development context, researchers have already argued that usability specialists should participate in OSS development (Bach and Carroll 2009, Bach et al. 2009, Andreasen et al. 2006, Benson et al. 2004, Cetin et al. 2007, Nichols and Twidale 2003, Nichols and Twidale 2006, Zhao and Deek 2005). However, usability specialists might not work in some OSS projects, and, even if they do, they might be very few in number and they might be working in isolation (Cetin et al. 2007, Nichols and Twidale 2003, Twidale and Nichols 2005). In addition, different kinds of usability evaluation methods have been suggested for specific activities. Several articles recommend conducting usability testing (Andreasen et al. 2006, Nichols and Twidale 2003, Nichols and Twidale 2006, Zhao and Deek 2005, Zhao and Deek 2006), while others argue for usability inspections as well (Zhao and Deek 2005, Zhao and Deek 2006). User interface design by blogs has also been suggested (Nichols and Twidale 2006), as has use of usability infrastructure such as discussion forums, mailing lists, and design areas (Bach et al. 2009, Bødker et al. 2007, Nichols and Twidale 2003, Nichols and Twidale 2006, Twidale and Nichols 2005).

Nevertheless, researchers have also argued that problems might arise when combining the traditional usability methods and recommendations with the OSS development model and philosophy (Bødker et al. 2007, Benson et al. 2004, Cetin et al. 2007, Nichols and Twidale 2003, Terry et al. 2010, Zhao and Deek 2005). A reluctance to heavyweight corporate usability processes might arise in OSS development; i.e., decentralized and engineering-driven OSS development might contrast with heavyweight corporate usability processes (Benson et al. 2004, Bødker et al. 2007, Cetin et al. 2007, Zhao and Deek 2005). In addition, usability specialists might have difficulty in showing their merits and gaining authority in OSS projects (Andreasen et al. 2006, Bach et al. 2009). They may be welcomed into the consultative role as feedback providers (cf. Iivari 2006b), but they might find it difficult to gain any decision-making power regarding the solution (Andreasen et al. 2006), which is recommended in a participative role (cf. Iivari 2006b). In OSS projects, the core developers typically make all of the decisions regarding what will be included in the code base (Ye and Kishida 2003). Nonetheless, usability activities should be made visible and merited so that usability specialists are able to gain more authority (Bach et al. 2009).

RESEARCH METHOD

This research effort can be characterized as design science research on exploring the ways of introducing usability activities from the traditional SW development context into the OSS development context. Design science research is about building artifacts for specific purposes and about evaluating how well they perform. Design science research involves a rigorous process to design artifacts—such as constructs, models, methods, or instantiations—to solve observed problems, to make research contributions, to evaluate designs, and to communicate the results to appropriate audience (Hevner et al. 2004, Peffers et al. 2007). Design science research can be seen as an embodiment of three closely related cycles of activities, namely Relevance cycle, Design cycle, and Rigor cycle

(Hevner 2007). The Relevance cycle inputs requirements from the contextual environment into the research and introduces the research artifacts into environmental field testing. The Design cycle supports a loop of research activity for the construction and evaluation of design artifacts and processes. The Rigor cycle provides grounding theories and methods, along with the domain experience and expertise from the foundations knowledge base into the research and adds the new knowledge generated by the research to the knowledge base (Hevner 2007, Peffers et al. 2007). The evaluation of artifacts considers the design fitness of the artifacts and their design utility on the environment and their users (Gill and Hevner 2011). In particular, we evaluate the design fitness and design utility of HCI approaches for introducing usability activities into software development in the OSS development context. These HCI approaches are the artifacts of this study. The evaluation of these HCI approaches has been carried out using four case studies. The results of these experiments have then been combined and contrasted with the existing HCI research knowledge related to introducing usability into software development, in order to generate new knowledge on the subject matter, particularly addressing this new context (i.e., OSS development). The research process has been iterative and incremental, and the design fitness and design utility of these HCI approaches has been evaluated after each iteration.

The case projects (Table 1) involved in this study were OSS development projects that were sufficiently large to ensure that the project would remain active during the period of usability team involvement. In practice, OSS development projects with just one or very few core developers can quite easily become inactive for uncertain periods of time, when one of the core developers is not active for some reason. Three of the projects were also small enough to allow easy identification and communication with the core developers. These OSS development projects involved development of software for nontechnical end-users. The third case project was much larger and involved development of software for users highly skilled in 3D content creation and computer use. The case projects were deliberately chosen from projects that did not involve software companies. This was because these companies may have usability resources that they utilize during development (e.g., Benson et al. 2004, Frishberg et al. 2002, Iivari et al. 2008, Nichols and Twidale 2006). The OSS projects that have been started and/or closely monitored by software companies might not be very different from proprietary software development, from the usability activities point of view. Therefore, these were not selected as cases.

Table 1: Description of the OSS Case Projects

	Case 1	Case 2	Case 3	Case 4
Number of developers	~30 developers	~15 developers	~40 developers	~20 developers
Number of users	Medium to large user base (<16.000)	Small user base (<1000)	Very large user base (>800.000)	Very large user base (>800.000)
Application type	Media center software	Game	3D content creation software	Media center software
Starting year	2004	2003	2002	2003
Development activity	Active development	Sporadic activity	Active development	Active development
Decision-making	Authoritative (core-developers)	Community based	Authoritative (core developers)	Authoritative (core-developers)
Usability team	5 students	3 students	10 students	3 students
Usability activities	Expert evaluation, usability testing	Expert evaluation, usability testing, user interface design	Expert evaluation, usability testing, mock ups	Expert evaluation, usability testing

Access to these four case studies was gained through four student projects, UKKOSS 1, UKKOSS 2, UKKOSS 3, and UKKOSS 4, which were aimed at introducing usability activities into OSS development projects under supervision of HCI researchers. The students conducting these usability projects had a usability background from at least two previous usability courses about usability evaluation methods (e.g., heuristic evaluation and usability testing), user-centered design, and user interface design in both theory and practice. Each student project consisted of three to five students working about 200 hours each in planning the usability activities, carrying out the usability activities, communicating with the OSS project, following up the impact of usability activities, collecting data, and writing a project report. The student projects introduced usability activities into one selected OSS development project, based on approaches and under the supervision and guidance of HCI researchers, and collected data

related to these usability activities and some issues related to the history, structure, and culture of the case OSS project. HCI researchers supervised and guided the usability activities introduced by the UKKOSS projects, analyzed the impact of usability activities in the case projects, and made plans for the following cases. Prior to these usability activities, the OSS case projects 1, 2, and 4 had very limited knowledge and background in usability activities. The OSS case project 3 had some knowledge of usability and user interface design, but very limited background in implementing usability activities.

The research material has been gathered over five semesters. The material was collected while conducting usability activities and observing OSS development projects. Usability findings and recommendations and all forum and email correspondence with the development teams have been saved for the purposes of the research. After each case study, the collected research material was analyzed and evidence related to the success of the selected approach was recorded.

EMPIRICAL CASES

The goal of the UKKOSS 1, UKKOSS 2, UKKOSS 3, and UKKOSS 4 projects was to determine the success of different approaches for introducing usability activities into OSS development projects when projects are offered usability help from outside. In none of these cases was the target OSS project commercially supported. Case projects 1 and 2 had a quite small developer group. Case projects 3 and 4 had a very large and active developer and community base. In each case, the student group conducted usability evaluations about the target software as a usability team and reported the results to the developers using different strategies.

UKKOSS 1—Consultative Approach

Approach 1: Usability activities should be introduced to OSS development by a consultative approach, in which external usability specialists provide feedback to the OSS solution. We reasoned that:

- OSS projects are meritocratic; one should be able to display her/his merits and abilities to others and gain recognition and status through them (Andreasen et al. 2006, Bach et al. 2009).
- Usability evaluations (empirical and expert evaluations) can function as a wake-up call that is needed when introducing usability into software development (Schaffer 2004).
- Empirical usability testing is the most widely utilized usability evaluation method (Gunther et al. 2001, Rosenbaum et al. 2000, Vredenburg et al. 2002). Arguments are also made for usability inspections in the literature (e.g., Mack and Nielsen 1994, Zhao and Deek 2005, Zhao and Deek 2006).
- Organizations should first hire external usability consultants for usability work before setting up an internal usability team (Schaffer 2004).
- Unsolicited usability feedback might be suitable approach, since usability activities can also be unsolicited in closed source software development. Project managers and developers might have no influence over when and what kinds of usability activities are performed in the project; e.g., in the case of usability testing or reviews acting as wake-up calls (cf. Shaffer 2004). Unsolicited usability feedback also reflects how the other contributions, such as code patches and additional features, are delivered to OSS projects.

In the first UKKOSS usability project, the target OSS project involved the development of media-center software. This project was chosen on the one hand because it was active and sufficiently large to be constantly active, but small enough to be approachable and, on the other hand, because the project identified the target users as ordinary people instead of experts. The project, initiated in 2004, had a total of about thirty developers (project leaders, committers, and contributors) and a medium-to-large user base of about 16,000 users. The goal of the usability team was to act as external usability consultants for this OSS project and to conduct expert evaluations and usability tests reporting the results from these to the OSS project without any previous communication. These usability activities were planned and conducted under supervision of a HCI researcher to ensure the good quality of these usability activities. Different kinds of usability evaluations were chosen based on selection of the most successful and widely used methods from the literature. Two types of expert evaluation were conducted for this software: heuristic evaluation with Shneiderman's heuristics and cognitive walkthrough (Shneiderman 1987, Wharton et al. 1992). Based on findings from expert evaluations, a usability test was planned (cf. Dumas and Redish 1993, Rubin 1994). Usability tests were conducted, with six users testing the software. Finally, the results were analyzed and a summary of findings was written as a report. The report included test objectives, test settings, tests tasks, positive feedback from test users, severity scale for usability issues, usability issues found in tests, and an overall summary. Usability issues listed in the report included a description of each issue, complete with screenshots, the user tasks where the issue was found, steps taken to replicate the issue, a severity rating for the usability issue, and proposed changes to the user interface to remove the issue.

This summary of usability findings was reviewed by the HCI researcher to ensure good quality and then was sent to the OSS project developers by email. This was the first contact between developers and the usability team, as planned. The purpose was to mimic the way that software patches are submitted to actual OSS projects (apart from the core developers): somebody writes a patch for themselves and then, once it is working for them, they share it with others and the core team can either accept the patch as part of the main branch or reject it. Based on the literature, we reasoned that having the usability activities fit into the OSS development context is important. This method also simulates the approach of bringing usability activities into traditional software development by hiring an external usability team that acts as consultants and carries out the usability activities in the consultative role using usability evaluation methods.

The results were not very encouraging. At first, no kind of answer or communication was received at all from the developers. Finally, the same document with the summary of the findings was posted to the project's website discussion forum. There, the developers answered that they were discussing this document internally and would comment on it later. After three years, no answer has yet arrived and there is no sign of any impact of these usability activities on future development and no sign of establishing an internal usability team. We planned to do a small-scale validation test with a similar OSS development project and similar strategy for introducing usability activities there. However, based on this case, just parachuting into OSS projects and sending the results of usability activities to OSS projects and expecting them to act did not appear to be the best course of action for those who want to introduce usability activities to OSS projects. This is the case, even though hiring external usability consultants has been advocated in the literature.

UKKOSS 2—Participative Approach in a Small OSS Project

Approach 2: Usability activities should be introduced to OSS development by a participative approach, in which usability specialists try to become recognized and appreciated by the core developers and by the whole community and are then allowed to take an active part in the design process along with the developers. We reasoned that:

- In addition to usability evaluations, usability specialists should be able to contribute more directly to design (e.g., Aucella 1997, Borgholm and Madsen 1999, Bødker and Buur 2002, Rosson and Carroll 2002).
- The developers should perceive usability specialists as teammates and allies (e.g., Aucella 1997, Bloomer and Croft 1997; Fellenz 1997, Mayhew 1999a, Rosenbaum et al. 2000, Tudor 1998).
- Executive usability championship is very important (Fellenz 1997, Schaffer 2004); in OSS development, the core developers might play a similar role (e.g., Aberdour 2007, Ye and Kishida 2003).
- Internal staff should be trained related to usability, and the usability team should act as evangelists, mentors, and trainers (Mayhew 1999, Schaffer 2004, Tudor 1998).
- Usability specialists should thoroughly understand the context into which the usability activities are to be introduced in order to select a suitable approach, since there is no "one size fits all"—a cultural match should be sought out (Aucella 1997, Iivari 2006a, Mayhew 1999a, Schaffer 2004).
- Usability cost-benefit analysis models outlining the potential cost and benefits of usability are important for selling usability activities to management (e.g., Rajanen and Iivari 2007). A refined usability cost-benefit model is needed for the OSS context, as the model is likely to be useful in selling usability activities to OSS development (Rajanen and Iivari 2010).

For the second OSS project, the plan was to perform similar usability activities, but with some changes regarding how to introduce them into the project. Again, a target OSS project that would be active, of suitable size, and targeted to ordinary people was chosen. The project, which started in 2003, had a total of about fifteen developers (project leaders, committers, and contributors) and a relatively small user base of about 1000 users. This time the chosen project was development of a game. Game usability and the quality of the user interface are very important for the players (Rajanen and Marghescu 2006). The most significant difference from the first case was that the developers were contacted right at the beginning, before conducting any usability activities. The goal was to become part of the community and to see if that would affect the developer's attitude toward the usability team and the work they were doing. The developers were told from the very beginning that the usability team was going to do some expert evaluations, and, after that, the usability team would provide any usability help the developers would ask for. This simulates how developers outside the core team become accepted members of the project community. The assumption was that the developers would be more willing to consider the input from the usability team if the team already had "sneaked" into the community and had an established identity within the community. The UKKOSS 2 usability team consisted of three students.

At first, contact with the developers consisted mainly of exchanging emails with the leading core developer of the project. In the beginning, the leading core developer was not even sure what usability meant, but the developer welcomed the help from the usability team. As the case went on, the usability team changed strategy, from sending documents and reports by email to the leading core developer, to chatting on the project's Internet Relay Chat (IRC) channel with the whole community about usability and introducing the potential benefits of better usability and risks of poor usability as outlined in the usability cost-benefit literature. On the whole, the help seemed to be appreciated and the awareness of the importance of usability rose in the community.

The usability team conducted heuristic evaluation to the game itself, combining several game heuristics (Desurvire et al. 2004, Federoff 2002). A similar report of usability findings and recommendations was written, as in the first case. The amount of usability findings and recommendations and their severity were similar, as in the first case. In addition, the usability team sent the developers an additional document in which they introduced themselves and described the background and goals of the usability team; this had not been done in the first case. The usability team also sent the developers some redesigned user interface mockups, which also has not been done in the first case. A summary of the findings, in addition to a more detailed report, was sent to the developers. The developers expressed interest in these documents. Later, these documents were placed on the development wiki, under "ToDo," so that anybody interested could read the suggestions and submit patches that would fix the issues that the usability team had identified. After the evaluation, the developers were asked what kind of work they would like the usability team to do. Because the developers had very little experience and knowledge about usability, they did not know what to ask. What they did know was that they wanted to improve their level editor so that anybody could pick it up quickly and easily, allowing them to have more user-submitted levels to their game. Based on these requirements, the usability team decided to do a cognitive walkthrough of the level editor and make sketches of a possible new user interface. Therefore, the team decided to take a more participative role in the development, as also recommended in the HCI literature (cf. Iivari 2006b).

Finally, the usability team conducted an interview in which the developers were asked how they felt about the usability activities, whether they found them useful, and, in their opinion, what things the usability team could have done better. This interview took place on the project's IRC channel so that all of the active members of the OSS project could express their views. The interview showed that these usability activities had improved awareness about the importance of usability and the benefits of better usability (acting as a wake-up call; cf. Schaffer 2004). The feedback from the usability team was welcomed, and it was taken seriously. Some mixed feelings were expressed about the level of communication. The developers still felt that they would have liked the usability team to have communicated with them more and to have been more active on the project's IRC channel. On the other hand, some of the developers thought that maybe keeping a certain distance helped with keeping an objective view. The developers would readily welcome any further usability help and hoped that they could receive it by having a usability specialist as a close-knit part of their team and the game development.

The developers also gave some negative feedback on the usability team's work. One of the main problems was the game's rapid development. Some of the usability team's findings were obsolete by the time the report was written because the developers were actively fixing bugs. It was clear that the usability team was treated as external consultants because the core developers did not help the usability team to fit the usability activities into their overall development and bug fixing plans. The usability activities were welcomed and encouraged, but the usability team did not become recognized members of the development community. The developers wanted concrete suggestions about how to fix the problems and not just general comments about what the problems were. Once the developers had a list of improvement suggestions, they would consider only the ones they saw as problems. If the core developers thought that some of the usability team's findings were not truly problems, these findings would get very low priority.

All in all, much more encouraging results were obtained from the UKKOSS 2 project than from the first UKKOSS project. In the case of Project 2, the core developers and contributors clearly were interested in users and in improving the usability of their software, but an apparent lack of knowledge about usability was evident and the developers wished for support from dedicated usability specialists. The core developers and contributors were interested in further collaboration, and we are planning to do a follow-up usability study with this project.

UKKOSS 3—Participative Approach in a Large OSS Project

Approach 3: Usability activities should be introduced to OSS development using a participative approach in which usability specialists try to become recognized and appreciated by the core developers and by the whole community. The more hierarchical structure of large OSS projects increases the difficulty in participating in the project in a way that recognition can be achieved. We reasoned that:

- Usability specialists should thoroughly understand the context into which the usability activities are to be introduced in order to select the most suitable approach, since there is no "one size fits all"—a cultural match should be sought out (Aucella 1997, Iivari 2006a, Mayhew 1999a, Schaffer 2004). Therefore, the size and hierarchical structure of the OSS project likely will have an impact on the selection of a suitable approach for introducing usability activities.
- Executive usability championship is very important (Fellenz 1997, Schaffer 2004). In OSS development, the core developers might play a similar role (e.g., Aberdour 2007, Ye and Kishida 2003). In larger and more hierarchical OSS projects, identifying and gaining access to core developers and getting executive usability champions might be more difficult.
- A wake-up call within core developers may be necessary in larger and more hierarchical OSS projects to gain access to these core developers and getting executive usability champion.

For the third OSS project, the plan was to perform similar usability activities, as in previous projects, but to try to infiltrate the usability team as part of the community from the very beginning. In order to ensure similarity and comparability of the usability activities with the previous projects, a member who had been in the UKKOSS 1 usability team and a project manager from the UKKOSS 2 usability team joined the usability team. This time, the case OSS project was large, with many core developers and a hierarchical, complicated structure with numerous active sub-communities based on locality or functionality, with their own sub-leaders, websites, and forums, etc. The software was very complex and aimed at users and user organizations having high-level domain expertise. The project, which started in 2002, had a total of about forty developers (project leaders, committers, and contributors) and a large user base of about 800,000 users. The goal of the usability team was to become part of the community and to see if the complex, hierarchical structure and challenging expert-user population (who had strong but differing opinions regarding how they wanted to use the software) would affect the ability of usability team to communicate with core developers and the developer's attitude toward the usability team and the work they were doing. The usability team familiarized themselves with the development history of the OSS project and followed development forums, IRC channels, and the wiki in order to identify the hierarchical structure of the project, to identify and gain access to core developers, and to identify potential usability champions among them. The team attempted to utilize usability evaluations as a wake-up call within the OSS project in a similar way to what would be done when introducing usability activities into traditional software development (cf. Schaffer 2004).

The case OSS project was the development of free open-source 3D content creation software with a vast number of content creation features and a goal of continually expanding its user base. The software was originally developed for in-house use in a commercial company before being released and further developed as open source software. New features were developed, one after another, by core developers and community members, but very few usability activities or user interface design were carried out beyond designing new icons for the interface. The usability team familiarized themselves with the OSS project by following project IRC channels, wiki, discussion forums, development history, "ToDo" lists, and roadmaps. Interesting findings related to usability or user interface discussions, activities, or arising usability benefit arguments were recorded. The team had some interesting findings regarding the culture and history of the case project, and it was surprised at how strong the core developers' opinions were in regard to the user interface. Some of the long time core developers had very firm opinions regarding how the user interface should compare with competitive commercial alternatives, meaning that the user interface must not be anything like the others. One of the core developers commented that usability was not something that would apply to this type of professional and complex system. However, another core developer noted that one of the goals of the project was to attract talented experts to use this open source system rather than similar commercial systems they were currently using at their work and in their free time. Attracting these types of high-class new users would bring prestige to the software and the OSS project developing it, much as increased usability could increase sales in closed source software development (cf. Rajanen and Iivari 2010). The software also had a notorious reputation, especially among new users, of being very difficult to learn and use, whereas more experienced users praised its efficiency after the steep learning curve was overcome. Substantial discussion occurred among developer and non-developer users in community forums and IRC channels about changes to be made to the user interface to make it easier to use. Constant peer support was available in community forums regarding problems in use, and the same issues were raised repeatedly. The users also made requests for features that already existed in the software, again indicating usability problems. The goals of attracting talented experts and reducing the reputation of the software as being very difficult to learn and to use were the key arguments used by the UKKOSS 3 project group to advocate for usability activities in the third case. The structure and size of the OSS project resembled a closed source software development project. The literature suggests that cost-benefit arguments can be used to gain an executive usability champion. The core developers in a large OSS development project resemble executives and managers in closed source projects. Thus, the argument can be made that usability cost-benefit arguments can be used in the OSS context to provide executive usability champions from core

developers, but these usability cost-benefit arguments need to be fitted to the OSS development context (Rajanen and Iivari 2010).

The UKKOSS 3 usability team consisted of two student teams. The first team of four students followed multiple community-based discussion forums and IRC rooms and promoted the usability work and results of the second team, while trying to gain an established identity within the core developers and community of the case OSS project. The second team of six students conducted all of the usability activities, including a full usability test with potential expert users. Finding suitable participants with talent and experience, but without previous knowledge of the case software, was very difficult; however, the team did manage to get four such potential expert users to test the software. Other usability activities, such as heuristic evaluation, were also performed, but the results were not found to be very practical in this case because the software was so complex and the widely used heuristics did not suit the 3D content creation software very well. The usability tests resulted in highly useful information about usability problems as well as superior aspects, when compared to similar commercial products that were familiar to the test users. All of the test users commented that they would certainly try this software again and that they quite likely would transfer to using this software instead of a commercial alternative at work or home, if the user interface were to be changed to be easier to use and if some conventions, like key shortcuts, would work or could be configured to work similarly to those in the commercial software they were using. These comments were also used as key arguments when promoting the results because these usability benefits matched with the aforementioned goals expressed by core-developers and the community.

Results from the usability tests and other usability activities were documented in open source fashion on a usability team website, and the website address was promoted in relevant community forums, IRC channels, and also offered to several community news sites. A similar report of usability findings, recommendations, and a redesigned user interface mockup, as in the second case, was made available on the usability team website. One core developer encouraged the UKKOSS 3 usability project and the usability activities, but did not take an active part in usability activities or their promotion, nor did this person become an active usability champion. This core developer ran the only community news website publishing news about these usability activities. This news site brought most of the hits to the usability project website, but at the end of the project, this community news website crashed and was down for many months, just at the most critical time when the usability project wanted to publish its results. Community discussion forums and IRC channels did not generate as much traffic to the usability project website because the news about usability activities was quickly buried among other discussions.

In this case, although the benefits of better usability to the OSS project were quite evident through improvement of the reputation of the software with respect to its difficulty to learn and use, and also through having more experienced users to switch from the commercial alternatives to this SW, the UKKOSS 3 usability project did not catch the attention of core developers and the community. Therefore, the usability team did not succeed in using the usability evaluations as a wake-up call within the OSS project in similar way as would occur when introducing usability activities into traditional software development (cf. Schaffer 2004). No further discussion related to UKKOSS 3 usability project occurred among the community or the core developers, and the results of the usability activities had no impact on development or on the user interface. Getting inside the community and becoming recognized as worthy contributors appeared to be a major hurdle in this particular OSS case because of the sheer size and complexity of the community and the complex hierarchical structure of the project.

The attention of both core developers and the community as a whole would likely have been easier to get had the UKKOSS 3 project been able to contribute to the user interface code and, in this way, gain an established position and identity within the project. Making user interface changes to software of this size, however, was beyond the possibilities of a student project. In addition, the tested version of the software did not allow for easy changes to user interface without also touching some core functionalities. In the current version of the software, released shortly after the student project ended, the user interface code is more separated from the functionality-related code, and, therefore, it would now be easier to modify the user interface.

One continuous problem for the UKKOSS 3 usability project was that a new software version was upcoming, which meant that the version they were testing and redesigning could already have been outdated when the first tests were made, but nothing certain was known about the new version at the time, except that some major changes to the user interface were expected. This problem of continuous change and redesign does not appear when doing usability activities in a normal closed source software development project with planned development cycles, activities, and roles. In OSS development, many different versions can exist under all kinds of states of development at the same time. Therefore, core developer involvement would seem to be necessary in OSS development so that the usability activities can be focused at the right time to the right version. We argue that usability activities can be focused at the right time to the right version when the OSS development project has a development plan or roadmap and when usability activities are planned together with the core developers.

All in all, the results from the UKKOSS 3 usability project were less encouraging than those from the second UKKOSS project. The third usability project team faced many challenges in getting their voice heard and did not succeed in becoming recognized community members.

UKKOSS 4—Consultative Approach

Approach 4: The results from the UKKOSS 1 case can be verified in other similar OSS projects. For the fourth OSS project, the plan was to repeat the usability tasks and phasing of the UKKOSS 1 case as closely as possible, with a similar type but different OSS project, to see if the results from the first case could be validated. The usability team had three students. As in the first case, the OSS project was development of media center software, the project was under active development, and the project's target users were ordinary people instead of technology savvy experts. The fourth OSS project was also large enough to be under constant development, but small enough so that the project leaders were identifiable and approachable. The project, which started in 2003, had about twenty active developers (project leaders, committers, and contributors) and a very large user base of more than 700,000 users. Expert evaluations were based on the project report of the first project so that they were similar, and usability testing was planned based on the results from the expert evaluations, as in the first project. Usability tests were conducted with four users testing the software, using similar tasks to those in the tests in first case, but slightly modified for testing this media center software. Finally, the results from the usability tests were analyzed together with the expert evaluation results, and a summary of findings was written as a report. The report positioned the student project as an external usability team for the project, described the types of expert evaluation and usability tests carried out, identified version of the software that was tested, gave a detailed description of the usability tests, including description of the test users and their tasks, reported findings from expert evaluation and usability tests, and gave improvement suggestions.

This summary of findings was then sent to the OSS project developers by email, as in UKKOSS 1. Similarly, this was the first contact between developers and the usability team, which mimicked the software patch submission process in OSS projects and, as mentioned in the first case, this approach simulated one way that the literature advises bringing usability activities into traditional software development, i.e., by an external usability team using discount usability methods.

Not surprisingly, the overall results were exactly the same as found for the UKKOSS 1 case. This time, the developers sent a brief reply acknowledging receipt of the usability report, but there was no further communication from the developers and the report was not further discussed or even mentioned in the OSS project discussion forums, chat, or mailing lists. Furthermore, no sign has appeared of any impact of these reported usability problems and suggested redesign solutions in the development plans of the project or in the user interface of the current version of the software. This small-scale validation test for the UKKOSS 1 indicates that merely sending the results of usability activities to OSS project and expecting them to act is not the best course of action for those who want to introduce usability activities to OSS projects, even though the literature has advocated this kind of approach in traditional software development. Usability activities alone did not result in a "wake-up call" (cf. Schaffer 2004) within the OSS project. Core developers and other contributors did not realize the importance of usability and no executive usability champion was gained for further promotion of usability activities.

DISCUSSION

Based on the four case studies we have carried out, we next discuss some findings in relation to the existing OSS and usability literature. We particularly highlight the importance of the usability specialists to partner with the OSS community and to gain core developer support. Practical advice for usability specialists is also offered.

Partnership with the OSS Community

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, since there is no "one size fits all" (Aucella 1997, Iivari 2006a). In the OSS context, the argument has been made that usability activities should be tailored to fit the OSS development philosophy and culture (Benson et al. 2004, Bødker et al. 2007, Cetin et al. 2007, Nichols and Twidale 2003, Terry et al. 2010, Zhao and Deek 2005). The developers have also been suggested as the most important target group, and they should perceive usability specialists as allies (Aucella 1997, Bloomer and Croft 1997, Fellenz 1997, Mayhew 1999a, Rosenbaum et al. 2000, Schaffer 2004). The point has also been raised that usability specialists are needed in OSS development, but their position tends to be quite challenging in OSS projects (Andreasen et al. 2006, Bach et al. 2009, Cetin et al. 2007, Nichols and Twidale 2003, Twidale and Nichols 2005).

These four empirical cases seem to suggest that the usability specialist should not be an outsider in OSS development, but should integrate with and become an active, visible member of the community. All four of the cases offer us evidence related to this, but case project 2 offers the most interesting evidence. Although the importance of the leading core developer having a personal interest and approval for usability activities in the second 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 the second case project, the OSS project was clearly interested in users and in improving the usability of their software, but a lack of knowledge also apparently existed regarding usability and potential benefits of usability. This supports the argument that the status of OSS usability is years behind the status of usability in the traditional software development, as reported in the literature (cf. Andreasen et al. 2006, Benson et al. 2004, Cetin et al. 2007, Nichols and Twidale 2003, Nichols and Twidale 2006, Zhao and Deek 2005, Zhao and Deek 2006). The second case project also showed that the usability activities performed by the usability team were a wake-up call for the developers (cf. Schaffer 2004). In the third case project, many core developers and community members had strong opinions about changing the user interface. Any suggestions regarding changing the user interface to resemble even slightly any of the competing commercial software interfaces resulted in bitter resistance and arguments, even if these changes would improve the usability. Consequently, this case points to the belief that the community as a whole should be committed to usability. In this case, the usability team did not succeed very well in becoming recognized members of the community, even though effort was devoted to this. However, one core developer published news related to their work, but due to some misfortunes, this did not happen at the most critical time and, therefore, was not adequate.

Based on the results of these four cases, we argue that the usability specialists should "infiltrate" into the development community, make themselves visible by providing information about usability and by evaluating and producing design solutions (cf. ISO 13407 1999), and offer their usability expertise to the core developers. These findings are in line with the findings by Bach and colleagues, who also mention educating the OSS community and building trust and community as ways of improving the position of usability (or user experience design) in OSS development (Bach and Carroll 2009, Back et al. 2009). Although the core developers may have a limited understanding of usability, the usability specialist should approach the core developers and contributors as peers rather than as pupils. Dialogue between usability team and development team is vital for the introduction of usability activities and for the usability activities to have any effect in the development. One should, overall, be able to adapt the usability activities into the development phase, culture, philosophy, and the vision of the core developers. This conclusion is connected to our findings from cases 2 and 3 related to the rapid development of the OSS and to the difficulty of targeting the usability activities to the right version at the right time. The usability activities need to be adapted to the way the rest of the development works, but at the same time, a certain distance also needs to be kept to maintain an objective view and the interests of the nontechnical end users in mind. This was brought up in the second case, where the developers thought that the usability team should keep a distance from them in order to maintain an objective view, but at the same time be a close-knit part of the development team, adapting to their culture and ways of work. The contradiction is apparent and raises very interesting challenges for adapting usability activities into OSS development. In OSS development, as well as in commercial settings, the challenge is for the usability specialists to align their ways of working with the engineers (Mayhew 1999a), but at the same time to preserve their role as the "representatives of nontechnical end-users" (cf. Iivari 2006a, Rajanen and Iivari 2007).

All in all, we argue that the usability specialists should adopt a participative role instead of a consultative role (cf. Damodaran 1996, Iivari 2006b) in the OSS development context and should try to make the developers their allies, in parallel with the findings from traditional software development projects (cf. Fellenz 1997, Mayhew 1999a, Rosenbaum et al. 2000). At first, a participative role might seem obviously better than parachuting straight into an unsuspecting OSS project as consultants from outside, but this should be seen as a benefit of hindsight, since the traditional usability literature also recommends the use of external usability consultants to start usability improvement, and in OSS projects, development input is typically gained in the form of unsolicited patches sent by contributors. Consequently, while the results from the first and fourth cases were not at all surprising, we wanted to test the effects and limits of this kind of suggested consultative approach, which would also have been a very cost-effective approach, the importance of which has also been highlighted in the literature (e.g., Bloomer and Croft 1997, Vredenburg et al. 2002). On the other hand, it needs to be highlighted that the participative role was not fully realized in our case studies, since the usability team did not have decision-making power regarding the OSS (cf. Damodaran 1996, Iivari 2006b). This probably is typical of OSS projects, since the core developers in OSS development normally make all of the decisions related to what to include in the OSS (Ye and Kishida 2003). Consequently, it would be highly surprising to observe the usability team to have gained decision-making power, especially in such a short timeframe. This will be discussed further in the following section.

Reconceptualizing Management Commitment for the OSS Development Context

Another finding from the traditional usability literature is the importance of management commitment and support when introducing usability into software development (Fellenz 1997, Schaffer 2004). Cost and benefit tradeoffs need

to be considered, and the management may be concerned with the possibility of usability activities increasing the development costs and time (Bloomer and Croft 1997, Vredenburg et al. 2002). This discussion bears interesting implications in the OSS development context, as, to a certain extent, the core developers can be argued to have the same characteristics as managers in the traditional software development. Core developers make both low level decisions regarding whether a particular contribution is accepted to the code or not, mid-level decisions regarding software features to be included into individual releases, and strategic decisions regarding the direction of the development in the future.

However, clear differences are also apparent. In the OSS development context, the core developers do not usually issue tasks for individual developers. Furthermore, in most OSS projects, the core developers do not have to budget and allocate limited human, technological, or financial resources. In traditional software development, the managers have to balance the development activities within the overall resources allocated to them. Since all usability activities require some kind of resources, management support is important when trying to bring usability activities to traditional software development. This is why the traditional usability cost-benefit analysis models focus heavily on gaining management commitment and support, by identifying areas of lower need for resources as a result of usability activities. This approach is not directly suitable for advocating usability activities in OSS projects, and, overall, the OSS context needs its own tailored usability cost-benefit aspects (Rajanen and Iivari 2010).

In the OSS development context, little justification is needed for the cost of resources to be used for usability activities because development is done on a voluntary basis and does not contain budgetary limitations. However, management support and commitment for usability activities in the OSS development context still need to be gained, and this may be accomplished through informing the core developers about usability and possible benefits of better usability for the project, through offering them usability resources and through planning with them how best to use the resources in the current phase of development. Hence, it is important that the usability team and the core development team are communicating from the beginning of usability activities.

Again, cases 2 and 3 offer us some evidence related to management commitment in OSS projects. Although, in the second case, most of the feedback was positive, not all of the suggestions made by the usability team were accepted. The core developers essentially considered these findings and suggestions and picked the ones they could agree with (in line with, e.g., Ye and Kishida 2003). If the developers did not think some issue was a problem, they would just ignore it and not include it in their “ToDo” list. This can be compared to the way patches are committed—if the core developers like the patch, it will get into the official release (cf. Aberdour 2007). If they do not like the patch, they can easily ignore it. In a similar way, the core developers can easily ignore usability activities if they do not know very much about usability in the first place. This was also apparent in the third case, where usability activities had a difficult time getting noticed without explicit core-developer support.

Interestingly, many potential benefits of better usability were brought up in the discussions of core developers and community in the third case project. (cf. Benson et al. 2004). As mentioned in the literature, it is important to be able to show the usability benefits that can be achieved (Mayhew 1999a, Rosenbaum et al. 2000). These kinds of considerations are especially important for management (Bloomer and Croft 1997) and likely for the core developers in the OSS development context. Indeed, some potential usability benefits were identified in the third case, including, for example, those related to gaining more users for the software if it would be made easier to learn and to having talented 3D content creators transfer from using similar commercial products if transfer from commercial 3D content creator software would be made easier by allowing the users to customize some key parts of the user interface and shortcuts to their liking. However, these identified usability benefits were not sufficient to result in a wake-up call for the need for usability activities among the core developers.

All in all, we argue that the project leader and core developers in OSS projects share some characteristics to management in proprietary software development in the sense of having decision-making power and influence on other members and project work. However, they may not have the same concerns with potential time and resource costs of usability activities faced by project management in proprietary software projects (cf. Rajanen and Iivari 2007), but neither seem to want usability activities to disturb the development flow. Therefore, the usability team must communicate closely with the core developers so that usability activities in OSS projects are done at the right time for the right software version without disturbing the overall development flow.

Student Involvement in OSS Research and Practice

We also recommend that others use student usability teams for introducing usability activities into OSS projects (see also Nichols and Twidale 2003). The use of the student teams to test different approaches for introducing usability activities into OSS projects and to gather data from them appeared to be a very useful and inexpensive, even though a time-consuming, method of research. When conducting this type of research with a student usability team in OSS, the timing should be planned for immediately after a release and the timeframe should be sufficiently long to

determine if the usability activities have an impact on the next release. This kind of planning gets complicated if the OSS project has already frozen the next release, in which case the possible impacts of the student usability team can be fully assessed only after a couple of releases. The student usability team should keep a close eye on all discussion forums and other sources of information regarding the target OSS project. We acknowledge that students acting as usability evaluators have limited skills when compared to professionals (see Connell and Hammond 1999, Desurvire et al. 1992, Skov and Stage 2003) and that this fact should be taken into account when selecting the students for the usability team and when supervising their work. The students in the usability team must have a good theoretical and practical understanding of the various usability analysis methods and design guidelines for user interface design, so that the results they produce are of high quality. In addition, the student usability team should also be under the close supervision of an experienced HCI researcher or teacher in the planning, implementation, and analysis phases, to ensure that usability activities are planned and followed through, and the results reported correctly and with high standards. The supervisor should review all usability activities, documentation, and communication between the usability team and the OSS project. The supervisor should also form a communication strategy regarding who communicates with the core developers and the community, what channel is used for the communication (e.g., discussion forum, email list), and what is communicated and when in relation to the goals of the research.

Nevertheless, student involvement as non-volunteer usability contributors can be seen as a limitation of this study that has to be taken into account when utilizing the results of this study. It should be taken into account that professional usability experts may have more credibility, and, therefore, their contributions may be valued as being more useful than less-experienced students. On the other hand, students from the information technology field actually act as fully fledged developers, contributors, and community members in many OSS projects, since these students have both development skills and time at their disposal. OSS projects do not require formal education from the developers, but instead value the ability to contribute something useful to the project regardless of the education or status of the contributor. Therefore, student usability contributors may actually have as much power as other student contributors within the community, although in these UKKOSS cases the student usability contributors were not volunteer contributors since they were asked to contribute as part of their studies. However, the usability contributors being students may not in itself lessen their power, since many of the developers may also be students, but it is a known problem that usability specialists in general have very little power in development projects, in the OSS context as well as elsewhere. It could actually be argued that the results could have remained the same even if professional usability experts had tried the same consultancy-style approach, because the OSS developer culture places more value of the functionality of the code than on interaction design (Green et al. 2009). Also, applying usability evaluation and UI design is best done prior to the start of SW development (Nichols et al. 2001, Nichols and Twidale 2003), which is usually impossible in OSS projects that have usually been initiated by technically oriented developers without any usability resources. Furthermore, OSS projects tend to be developed piece by piece by separate individual contributors, and, therefore, comprehensive design and evaluation activities do not fit this development process well (Green et al. 2009). Despite these difficulties, we argue that student usability teams can be used for introducing usability activities into OSS projects if the students are selected carefully and if an experienced HCI researcher or teacher closely supervises their work.

CONCLUSIONS

This paper examined the question: “How to introduce usability activities into OSS development?” The main findings, based on the four empirical studies as well as relevant usability literature, can be summarized as follows. We suggest that to bring usability activities successfully into an OSS project, the usability specialists should adopt a participative role instead of the consultative role, and:

- Understand the philosophy, principles, and characteristics of OSS development
- Initiate a peer-to-peer dialogue with the core developers and endeavor to make them allies
- Identify potential benefits of better usability and use them as arguments for usability activities
- Adapt the usability activities to the development, but maintain an objective view
- Keep the core developers and the community informed about usability activities and suggested user interface improvements
- Be aware of constant change and redesign, to carry out the usability activities for the right version of the software a the right time of the development
- Always remember to promote the interests of the nontechnical users

Many highly practical findings were identified in this paper for people interested in introducing usability activities into OSS projects. As a limitation of this study, the usability teams were not involved with the case projects from the

beginning of these projects. This was due to the long time necessary for an OSS project to evolve into a thriving community-supported development project. Usability literature argues that usability specialists should take part from the beginning of the development process. To be able to take part from the beginning of an OSS development project, the usability specialist should already have an established role in task and interface design when the project begins and no software has yet been produced. This would require the OSS developer initiating the project to have substantial knowledge and appreciation of usability to begin with. The OSS development philosophy of the "perpetual beta," where the software is constantly under redevelopment, is also a challenge for traditional usability activities and processes. As already mentioned, the software version that the usability team was testing in the second and third case was already old when evaluations were just beginning, and, therefore, the usability team was constantly trying to shoot a moving target with their expert evaluations and improvement suggestions. The usability teams also consisted of students, which is a limitation of this study. However, we argue that this did not necessarily have a significant impact on the results because students from this field are involved in OSS development projects anyway, as developers and community members, and the students were under the close supervision of the HCI researchers.

Interesting future areas of research would include determining how the effectiveness of usability work would change if the specialists were even more integrated into the community. The potential risk of usability specialists losing their objective view as a result of this close integration with development when acting more in participative than consultative role should also be examined. Determining the effect of the size of the project on the easiness and acceptance of the usability activities would also be interesting. Another path for future work would be to study if "infiltration" with usability activities is more difficult into a tight OSS community hierarchy than into an OSS community with less hierarchy. Interesting similarities and differences also are evident between OSS project leaders and managers of traditional software development projects that could be studied further. Further research into the possibilities of using the potential benefits of better usability as an argument for bringing usability activities into OSS project would also be interesting. The methods, the level, and the difficulties raising usability in OSS development up from grassroots activity into more organized and institutionalized activity should also be researched. A more action-research-oriented approach could be utilized in prospective studies in the OSS context. Finally, one could study further how far the usability activities and processes can be adapted into OSS development philosophy without compromising the core HCI philosophies and how OSS development practices could be adapted to enable easier incorporation of usability activities without compromising the core OSS development philosophies. We have done a validation study with a similar OSS development project as UKKOSS 2 and with a similar approach for introducing usability. The usability team conducted testing and redesign, while the code team implemented the changes to some parts of the user interface and communicated the necessities of these changes to the core-developers and community. Our initial finding from this study is that the code team can more readily establish themselves within the community and communicate usability problems, needs for change, and benefits of better usability to the core-developers by backing up their arguments with an already working code and, in this way, gain prestige among their peers.

REFERENCES

Aberdour, M., "Achieving quality in open source software," *IEEE Software,* 2007, 24:1, pp. 58–64.

Agerfalk, P., and B. Fitzgerald, "Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy," *MIS Quarterly,* 2008, 32:2, pp. 385–409.

Allen, C.D., "Succeeding as a clandestine change agent," *Communications of the ACM,* 1995, 38:5, pp. 81–86.

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.

Apache.org, "About," 2011. Available at: http://httpd.apache.org, last accessed 18 January 2012.

Aucella, A., "Ensuring success with usability engineering," *Interactions,* 1997, May + June, pp. 19–22.

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.

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.

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

Bevan N., "*Cost Benefit Analysis,*" TRUMP report, 2000, September.

Beyer, H., and K. Holtzblatt, *Contextual Design: Defining Customer-centered Systems,* San Francisco: Morgan Kaufmann Publishers Inc, 1998.

Bloomer, S., and R. Croft, "Pitching usability to your organization," *Interactions,* 1997, Nov. + Dec., pp. 18–26.

Borgholm, T., and K.H. Madsen, "Cooperative usability practices," *Communications of the ACM,* 1999, 42:5, pp. 91–97.

Bødker, S., and J. Buur, "The design collaboratorium: A place for usability design," *ACM Transactions on Computer-Human Interaction (TOCHI),* 2002, 9:2, pp. 152–169.

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,* Aykin, N. (ed.), LNCS 4559, 2007, pp. 10–18.

Cajander, Å., J. Gulliksen, and I. Boivie, "Management perspective on usability in public authority—A case study," In *Proc. of NordiCHI*, 2006, Oslo, Norway, pp. 38–47.

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*, Schuler, D. (ed.), LNCS 4564, 2007, pp. 32–40.

Connell, I.W., and Hammond, N. V., "Comparing usability evaluation principles with heuristics: Problem instances vs. problem types," In *Proc. INTERACT'99,* Sasse, M.A., and C. Johnson (eds.), 1999, pp. 621–629.

Cooper, A., and R. Reimann, *About Face 2.0: The Essentials of Interaction Design,* Hoboken, NJ: Wiley, 2003.

Dahlander, L., and M. Magnusson, "Relationships between open source software companies and communities: Observations from Nordic firms," *Research Policy,* 2005, 34, pp. 481–493.

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

Davis, F., "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, 1989, 13:3, pp. 319–340.

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.

Desurvire, H., J. Kondziela, and M. Atwood, "What is gained and lost when using evaluation methods other than empirical testing," In *Proc. HCI'92*, Monk, A.M., D. Diaper, and M.D. Harrison (eds.), 1992.

Donahue, G., "Usability and the bottom line," *IEEE Software*, 2001, 18:1, pp. 31–37.

Dumas, J., and J. Redish, *A Practical Guide to Usability Testing*, Norwood: Ablex Publishing Corporation, 1993.

Ehrlich, K., and J. Rohn, "Cost justification of usability engineering: A vendor's perspective," In *Cost-Justifying Usability*, Bias, R., and D. Mayhew (eds.), Waltham, MA: Academic Press, 1994, pp. 73–110.

Federoff, M., "Heuristics and usability guidelines for the creation and evaluation of fun in video games," 2002. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.8294&rep=rep1&type=pdf, last accessed 18 January 2012.

Fellenz, C.B., "Introducing usability into smaller organizations," *Interactions*, 1997, 4:5, pp. 29–33.

Fitzgerald, B., "The transformation of open source software," *MISQ,* 2006, 30:3, pp. 587–598.

Fitzgerald, B., and P. Ågerfalk, "The mysteries of open source software: Black and white and red all over?" In *Proc. of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*—Track 7, 2005, vol. 7, p. 196a.

Frishberg, N., A.M. Dirks, C. Benson, S. Nickel, and S. Smith, "Getting to know you: Open source development meets usability," In *Extended Abstracts of the CHI 2002,* 2002, pp. 932–933.

Gill, T.G., and A.R. Hevner, "A fitness-utility model for design science research," *Lecture Notes in Computer Science*, 6629, pp. 237–252.

Giuri, P., M. Ploner, F. Rullani, and S. Torrisi, *Skills and Openness of OSS Projects: Implications for Performance* (Working Paper), Laboratory of Economics and Management, Sant'Anna School of Advanced Studies, Pisa, Italy, 2004.

Green, C., A. Eiser, I. Tollinger, E. Castro, C. Ratterman, A. Vera, and G. Pyrzak, "Leveraging open-source software in the design and development process," In *Proc. 27th International Conference Extended Abstracts on Human Factors in Computing Systems (CHI 2009)*, 2009, pp. 3061–3074.

Grudin, J., "Systematic sources of suboptimal interface design in large product development organizations," *Human-Computer Interaction*, 1991, 6:2, pp. 147–196.

Gulliksen, J., B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and Å Cajander, "Key principles for user-centred system design," *Behaviour & Information Technology,* 2003, 22:6, pp. 397–409.

Gunther, R., J. Janis, and S. Butler, "The UCD decision matrix: How, when and where to sell user-centered design into the development cycle," 2001. Available at: http://www.ovostudios.com/upa2001/, last accessed 18 January 2012.

Hevner, A.R., "The three cycle view of design science research," *Scandinavian Journal of Information Systems*, 2007, 19:2, pp. 87–92.

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

Hutchings, A.F., and S.T. Knox, "Creating products—customer demand," *Communications of the ACM*, 1995, 38:5, pp. 72–80.

Iivari, N., "Discourses on 'culture' and 'usability work' in software product development," *Acta Universitatis Ouluensis, Series A, Scientiae rerum naturalium 457*, 2006a.

Iivari, N., "Understanding the work of an HCI practitioner," In *Proc. 4th Nordic Conference on Human Computer Interaction,* Morch, A., K. Morgan, T. Bratteig, G. Ghosh, D. Svanaes (eds.), 2006b, October 14–18, pp. 185–194.

Iivari, N., H. Hedberg, and T. Kirves, "Usability in company open source software context: Initial findings from an empirical case study," In *Proc. 4th International Conference on Open Source Systems* (co-located with *IFIP 20th World Computer Congress),* Russo, B., E. Damiani, S. Hissam, B. Lundell, G. Succi (eds.), 2008, pp. 359–365.

ISO 13407, "Human-centered design processes for interactive systems," *International standard,* 1999.

ISO/IEC 9126-1, "Software engineering, product quality, Part 1: Quality model," *International Standard,* 2001.

Karat, C.M., "A business case approach to usability cost justification," In *Cost-Justifying Usability,* Bias, R., D. Mayhew (eds.), Academic Press, 1994, pp. 45–70.

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

Klein, H.K., and M.D. Myers, "A set of principles for conducting and evaluating interpretive field studies in information systems," *MIS Quarterly*, 1999, 23:1, pp. 67–94.

Lakhani, K., and E. von Hippel, "How open source software works: 'Free' user-to-user assistance," *Research Policy, 2003,* 32, pp. 923–943.

Mack, R., and J. Nielsen, "Executive summary," In *Usability Inspection Methods*, Nielsen, J., and R. Mack (eds.) Hoboken, NJ: John Wiley & Sons, 1994, pp. 1–23.

March, S.T., and G.F. Smith, "Design and natural science research on information technology," *Decision Support Systems*, 1995, 15:4, pp. 251–266.

Mauro, C., "Usability science: Tactical and strategic cost justifications in large corporate applications," In *Cost-Justifying Usability: An Update for an Internet Age,"* Bias, R., and D. Mayhew (eds.), Waltham, MA: Morgan Kaufmann, 2005, pp. 265–296.

Mayhew, D., "Strategic development of usability engineering function," *Interactions*, 1999a, 6:5, pp. 27–34.

Mayhew, D., *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*, Waltham, MA: Morgan Kaufmann Publishers, 1999b.

Mayhew, D., and M. Mantei, "A basic framework for cost-justifying usability engineering," In *Cost-Justifying Usability*, Bias, R., and D. Mayhew (eds.), Waltham, MA: Academic Press, 1994, pp. 9–43.

Mozilla.com, "About," 2012. Available at: http://www.mozilla.com, last accessed 18 January 2012.

Nichols, D., K. Thomson, and S.A. Yeates, "Usability and open source software development," *Proc. of the Symposium on Computer Human Interaction* (SIGCHI New Zealand), 2001, pp. 49–54.

Nichols, D., and M. Twidale, "The usability of open source software," *First Monday,* 2003, 8:1.

Nichols, D., and M. Twidale, "Usability processes in open source projects," *Software Process Improvement and Practice,* 2006, 11, pp. 149–162.

Niederman, F., A. Davis, M. Greiner, D. Wynn, and P. York, "A research agenda for studying open source I: A multilevel framework," *Communication of the Association for Information Systems,* 2006, 18, pp. 19–149.

Nielsen, J., *Usability Engineering,* Academic Press, Boston, 1993.

Ohnemus, K., "Incorporating human factors in the system development life cycle: Marketing and management approaches," In *Proc. IEEE IPCC96*, 1996, pp. 46–53.

OpenOffice.org, "About," 2012. Available at: http://www.openoffice.org, last accessed 18 January 2012.

Peffers, K., T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, 2007, 24:3, pp. 45–77.

Rajanen, M., "Different appraches to usability cost-benefit analysis," In *Proc. ECITE 2006,* Remenyi, D., and A. Brown (eds.), Academic Conferences, Reading, PA, 2006, pp. 391–397.

Rajanen, M., and N. Iivari, "Usability cost-benefit analysis: How usability became a curse word?" In *Proc of INTERACT 2007: 11th IFIP TC13 International Conference on Human–Computer Interaction,* Baranauskas, C., P. Palanque, J. Abascal, and S. Barbosa (eds.), Lecture Notes in Computer Science 4663, 2007, pp. 511–524.

Rajanen, M., and N. Iivari, "Traditional usability costs and benefits—Fitting them into open source software development," In *Proc. 18th European Conference on Information Systems (ECIS 2010)*, 2010.

Rajanen, M., and D. Marghescu, "The impact of game usability to player attitude," In *Proc. of 29th Information Systems Research Seminar in Scandinavia (IRIS29)*, Helsingoer, Denmark, 2006.

Raymond, E., *The Cathedral & the Bazaar: Musing on Linux and Open Source by an Accidental Revolutionary*, Sebastopol, CA: O'Reilly Media, 1999.

Rogers, E., *Diffusion of Innovation, 5th Edition*," New York: Free Press, 1995.

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., G. Szwillus, M. Czerwinski, F. Paterno, S. Pemberton (eds.), ACM, New York, 2000, pp. 337–344.

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

Rubin, J., *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Test*, Hoboken, NJ: John Wiley & Sons, 1994.

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

Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Boston: Addison-Wesley, 1987.

Skov, M.B., and J. Stage, "Supporting novices in detecting Web site usability problems: A comparison of the think-aloud and questionnaire methods," In *Proc. of the 10th Conference on Human–Computer Interaction*, 2003, pp. 858–862.

SourceForge.net, "About," 2012. Available at: http://sourceforge.net, last accessed 18 January 2012.

Tudor, L., "Human factors: Does your management hear you?" *Interactions*, 1998, Jan. + Feb., pp. 16–24.

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

Vredenburg, K., J. Mao, P.W. Smith, and T. Casey, "A survey of user-centered design practice," In *Proc. of CHI'02,* Wixon, D. (ed.), ACM, New York, 2002, pp. 471–478.

Wharton, C., J. Bradford, J. Jeffries, and M. Franzke, "Applying cognitive walkthrough to more complex user interfaces: Experiences, issues and recommendations," In *Proc. CHI 1992*, pp. 381–388.

Ye, Y., and K. Kishida, "Toward an understanding of the motivation of open source software developers," In *Proc. of 25th International Conference on Software Engineering (ICSE),* IEEE, 2003, pp. 419–429.

Zhao, L., and F. Deek, "Improving open source software usability," In *Proc. of the 11th Americas Conference on Information Systems,* 2005, pp. 923–928.

Zhao, L., and F. Deek, "Exploratory inspection: A learning model for improving open source software usability," In *Extended Abstracts of the CHI2006,* 2006, pp. 1589–1594.

ABOUT THE AUTHORS

Mikko Rajanen is a teaching assistant in the Department of Information Processing Science at the University of Oulu, Finland. He received his Ph.D. in 2011 after defending his thesis "Applying Usability Cost-Benefit Analysis—Exploration in Commercial and Open Source Software Development Contexts." His research interests include introducing usability activities into open source software development, usability in computer games and usability cost-benefit analysis.

Dr. **Netta Iivari** holds the position of a Post-Doctoral Research Fellow from the Academy of Finland. She is an Adjunct Professor at the University of Oulu with the field "user-centered, participative information systems design." She received her doctoral degree in Information Systems from University of Oulu. She also has a master's degree in Cultural Anthropology from University of Jyväskylä. Her research is strongly influenced by interpretive and critical research traditions. Her long-lasting research interest is related to participation of different kinds of stakeholder groups in defining, designing, and shaping information technologies in their everyday life. Her empirical work has been carried out in packaged and open source software development contexts and related to solutions targeted at children. She has published in journals such as *Information Systems Journal, Behaviour and Information Technology, Information & Organization, Journal of Organizational and End User Computing, Interacting with Computers,* and *Information Technology & People.*

Kaisa Anttila is a student in Department of Information Processing Science in University of Oulu, Finland. She participated in all the UKKOSS-projects described in this paper either as one of the students conducting the usability activities or as an expert of previous projects. Her research interests include usability, open source software development, and social media.

JITTA

JOURNAL OF INFORMATION TECHNOLOGY THEORY AND APPLICATION

JITTA IS A PUBLICATION OF THE ASSOCIATION FOR INFORMATION SYSTEMS
ISSN: 1532-3416

Volume 12 ■ Issue 4 ■ Article 2