Rajanen, M., Iivari, N., Keskitalo, E. (2012) Introducing Usability Activities into Open Source Software Development Projects - a Participative Approach. In Proceedings of the 7th Nordic Conference on Human-Computer Interaction (NordiCHI 2012). Copenhagen, Denmark. pp 683-692. ISBN: 978-1-4503-1482-4, DOI: 10.1145/2399016.2399120

Introducing Usability Activities into Open Source Software Development Projects – a Participative Approach

Mikko Rajanen, Netta Iivari and Eino Keskitalo
Department of Information Processing Science
University of Oulu
P.O. Box 3000, 90014, Oulu, Finland
mikko.rajanen@oulu.fi, netta.iivari@oulu.fi, eino.keskitalo@gmail.com

ABSTRACT
Usability is an important quality characteristic of software products and information systems. Different approaches for introducing usability activities into open source software (OSS) development have not yet been fully evaluated. This paper experiments with the introduction of usability activities into OSS development through a participative approach. An empirical case study was carried out in a game development OSS project. The results of this study suggest that it is beneficial to introduce usability activities into OSS development through the participative approach. In the participative approach the usability experts become recognized part of the development community through adapting their ways of work into the culture of the OSS project and submitting code patches. This participative approach had a clear impact in the case project as seen in changes in the user interface and in improved usability. The challenge of adapting usability and OSS development philosophies and practices should, however, be researched further.

Author Keywords
Usability; open source software; participative approach

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

This paper presents an empirical case study where usability activities are introduced into open source software (OSS) development context. OSS development has already been acknowledged as a current and interesting topic in Human Computer Interaction (HCI) research (e.g. [5, 8, 9, 31, 32, 46, 48]), OSS development referring to the development of software, whose source code needs to be open for anyone to use, modify and distribute (e.g. [27, 44, 20]). The development model relies on technically skilled individuals who develop the software to serve their personal needs, but also voluntarily allow the software to be used and further developed by others. OSS projects rely on community software development model that extensively utilizes Internet's means for communication and for coordinating the work: discussion forums, mailing lists, IRC, version control and bug reporting systems are widely used in OSS development projects. [27, 44, 20]

NordiCHI '12, October 14-17, 2012 Copenhagen, Denmark

OSS development has already become a highly significant phenomenon. It is difficult to estimate the number of users and overall market share of OSS solutions because they can be downloaded freely from mirror repositories and peer-to-peer networks, but the significance of OSS development becomes evident when, for instance, looking at SourceForge, one of the Web-based code repositories used by OSS development projects. SourceForge alone has about 2.7 million developers, more than two million downloads from its repositories every day and the combined number of users in all projects is estimated 46 million. There are at least twenty similar code repositories. Regarding the usage of OSS solutions, for instance the Firefox Web browser has been downloaded over one billion times and it has a 23 percent worldwide usage share.

Even though OSS development originates in the hacker culture with highly talented developers [27, 20], recently also non-developer users have emerged in the OSS scene. These users require high level usability in addition to advanced functionality. Traditionally in OSS development usability of the software has not been a major concern, but nowadays it needs to be considered in connection to many OSS solutions – those that try to please also the non-developer user population – but the introduction of usability into OSS development has proven to be quite challenging. [1, 8, 31, 32, 44]. For these reasons there is a pressing need for studies of this kind – tackling the problem of introducing usability activities into OSS development.

The overall research question examined in this paper is: "How could usability activities be introduced into OSS development?" The specific research question of this paper is: "How could usability experts participate within OSS community in order to usability activities have an impact?" This paper contributes by experimenting with a participative approach (see [22, 40]) for introducing usability activities into software development in the OSS development context. This approach has already been

recommended for the OSS development context [40], but its implementation so far has had certain limitations that will be addressed in this study. There is a lack of research of this kind, but it is evidently needed, as the existing research has shown that more work needs to be done related to improving usability in the OSS development context and that introducing usability activities into OSS development is challenging (e.g. [8, 9, 31, 32, 46, 47]).

The paper is structured as follows. The next section introduces the concept of usability and outlines some example methods devised for building and ensuring it. The next section also discusses the existing research results related to introducing usability activities into traditional, commercial software development context as well as to the OSS development context. The following section discusses the research method utilized in this study, the fourth section outlining the empirical results gained. The fifth section discusses the implications of the results, the final section summarizing the main results, outlining their limitations and implications for practice and identifying paths for future work.

USABILITY IN SOFTWARE DEVELOPMENT

Definition and design of usability

Usability is identified as one of the main software product and system quality attributes in the international standard ISO 1926. It has been discussed particularly with respect to HCI research which has also introduced a number of different methods for improving and designing usability (e.g. [29, 32, 38]). Usability 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 for specified tasks [24]. International 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" [23].

Usability has become an important competitive edge in software markets long ago, i.e. top class functionality has not been sufficient anymore, but customers have demanded also usability to be on the high level [16, 33, 41]. Users or user organizations can benefit from better usability through higher user satisfaction as well as through higher productivity when the most frequent tasks take less time and users make fewer errors [13, 30]. 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 [13, 6, 30]. Furthermore, the development company can use improved usability as a competitive edge and potentially increase sales [13, 6, 26, 30]. If the development company provides some kind of product support, better usability can also greatly reduce the number of contacts made to customer support and reduce support costs [13, 6, 30]. As can be seen, there are numerous motivations for the development organization (or community) to invest in usability.

Usability can be designed and improved through usability engineering (UE) and user-centered design (UCD) methods. A number of such methods have been developed within HCI research (e.g. [23, 29, 33, 41]. For example, usability engineering (e.g., the original work in [33], also e.g. 29]), scenario-based design (e.g., [41]) and goal-based interaction design (e.g. [10]) methods have been developed that contain usability activity phases such as requirements analysis, activity design, information design, interaction design, prototyping, usability evaluation, and documentation design [41] or research, modeling, requirements definition, framework definition, and refinement [10].

Introducing Usability into Software Development

Even though there are a plenty of methods for ensuring usability, there still is research lacking on how to introduce these methods, and usability activities in general, to the development. The introduction of usability activities even into traditional software development has been reported as challenging – already decades ago as well as currently (see e.g. [17, 22, 34, 38]). A number of reasons can be outlined in relation to this. For example, cost benefit considerations and resource constraints have been emphasized as reasons. Therefore, resources required for usability activities need to be well planned and budgeted [3, 28, 33] and one should ensure that usability activities do not increase development costs and time [7, 17, 33, 38]. The existing research has revealed that usability may not be appreciated as an important factor by developers and their managers and usability experts may have difficulties in becoming visible and legitimate participants in the development process, in gaining decision-making power regarding the solution, and in their work, altogether, having any impact on the solution (see e.g. [3, 17, 22, 28). Furthermore, the context into which the usability activities are to be introduced needs to be thoroughly understood in order to select a suitable strategy, as a 'one size fits all' approach is not recommended but instead a 'culturally compatible' one [2, 3, 9, 22, 28, 37, 45].

The introduction of usability activities into OSS development is challenging as well. Traditionally, technically skilled developers have developed OSS for their own use, but now OSS solutions have increasing amount of users who lack deep technical knowledge or skills to contribute to the OSS projects by submitting code patches. Recently, interest in usability research in the OSS development context has emerged (e.g. [1, 39, 8, 9, 46]). The current status of usability activities in OSS projects and usability of OSS is still generally quite poor and the usability experts, if any, may work in isolation their work having no impact on the actual solution. These problems are particularly present in small and medium sized OSS

projects and in OSS projects without company involvement. (See e.g. [1, 9, 31, 32, 46, 47].)

Particularly, there still is a lack of research on how to introduce usability activities into the OSS development context so that those have impact on the actual solution. One study has explicitly experimented with different approaches for introducing usability activities into OSS development (see [40]). This study suggests that usability experts should adopt a *participative approach* in introducing the usability activities, i.e. they should become recognized members of the OSS community, understand the principles and the culture of the community, adapt the usability activities to the development and keep the core developers and the community informed about the usability activities [40]. The results are in line with studies carried out in the traditional software development context that maintain that usability specialists should become team members and allies in the development, align their ways of work with the engineers, take care that usability becomes a visible and legitimate part of the development and be wary that the usability experts do not become viewed as police pointing out only negative issues through their usability evaluations [3, 17, 22, 28, 45]. Also the few other studies addressing the helping and hindering factors related to the entrance of usability activities in OSS development provide support for these claims (i.e. [4, 5, 8, 9]).

The advice provided (in [40]), therefore, is likely valid, but it is problematic that this study did not go very far with the participative approach. This is because although usability problems were identified and mock-ups for improving user interface were produced, the usability teams did not submit patches and they were involved with their respective OSS projects for only a few months. The usability experts got to know the OSS community in question through reviewing all the relevant communication (i.e. in the discussion forums, mailing lists, IRC) and through directly contacting the (preferably core) developers and the community more generally. In addition, they provided not only usability feedback, but also improvement suggestions, and actively tried to inform the community of their work. (See [40]). There were, however, clear limitations regarding the implementation of this approach. As has been suggested by the HCI research addressing the traditional, commercial software development context (e.g. [2, 3, 17, 21, 22, 37, 45]) as well as the OSS development context [1, 4, 5, 8, 9, 31, 46, 47), the usability experts should become fellow members in the development, build trust and show their merits, align their ways of work with those of the developers' and select an approach that is 'culturally compatible'. Therefore, we reasoned that usability experts should become even more tightly integrated with the development: to enter the community and to gain merit in a similar way than other to-be community members, i.e. through submitting patches. As contributors, the usability experts would be entering and contributing to in this development context clearly in a 'culturally compatible' way. By this, we do not mean, however, that all usability experts should be capable of coding to be able to enter OSS projects, but we still maintain that there should be at least a person in mind that is known to be capable and willing of implementing the usability improvements of the usability team when entering an OSS project.

RESEARCH METHOD

This research is characterized as design science research because it explores the introduction of usability activities into the OSS development context –first by devising a way related to how to introduce the usability activities into OSS development based on the existing literature on the matter and afterwards by experimenting with it in practice in a selected OSS development project. Design science research is about building artifacts for specific purposes, and about evaluating how well they perform in their intended purposes. Design science research involves a rigorous iterative and incremental 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 [19, 35]. In this case a method (or an early version of such) for introducing usability activities into OSS development has been under construction and evaluation. Based on a literature review, a set of usability activities was selected with the aim to introduce them into OSS development. However, the specific research interest was in the actual procedure related to how to introduce those activities into the selected OSS development project. This procedure was also based on the literature on the matter, combining general HCI literature on the introduction of usability activities into software development with the literature addressing specifically the OSS development context. In particular, the participative approach for introducing usability activities into software development in the OSS development context was the artifact of this study under construction and evaluation.

Design science research can be seen as an embodiment of three closely related cycles of activities, namely Relevance cycle, Design cycle and Rigor cycle [18]. The Relevance cycle inputs requirements from the contextual environment into the research (i.e. OSS development context) and introduces the research artifacts (i.e. ways to introduce usability into OSS development context) into environmental field testing. The Design cycle involves the actual construction and evaluation of the 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 [18, 35]. The evaluation of artifacts considers the design fitness of the artifacts, and their design utility on the environment and their users [14]. In particular, we evaluate the design fitness and design utility of the participative approach for introducing usability activities

into software development in the OSS development context. The evaluation has been carried out in a long-term case study by using student usability teams working with one OSS development project. The results of this case study have then been contrasted with the existing HCI research knowledge related to introducing usability activities into software development with the aim to generate new research knowledge on the subject matter, addressing particularly this new context, i.e. OSS development. The design fitness and design utility of this participative approach have been evaluated after the student usability teams have finished their work.

The research material used in this study has been collected while conducting the usability activities and observing the selected case OSS development project. The student usability teams introduced the usability activities into the case OSS development project and collected data related to the usability activities and related to the case OSS development project. Experienced HCI researchers supervised and guided the usability activities as well as analyzed the impact of the usability activities in the case OSS development project. Usability findings and recommendations and all forum and email correspondence with the development team have been saved for the purposes of the research. The collected research material has been analyzed by the experienced HCI researchers; evidence related to the successes and shortcomings of the selected approach was recorded and the findings were contrasted with the existing HCI research knowledge.

EMPIRICAL FINDINGS

The Case Project

The case project involved in this study is an OSS development project developing a single-player, role-playing roguelike game with both ASCII and GUI user interfaces. The development of the game originally started in 1995. The development team has changed many times since then and the development has branched into several separate development projects. This OSS project was under active development during the two years of usability student team involvement. It had twenty active core developers with commit rights and several other contributors developing around 7500 commits to the project repository, an active community including a forum with over 1000 users and around 50000 posts, and a generally open and friendly culture welcoming all those who are willing to somehow contribute to the project or were interested in the game.

The case project is not commercially supported. We intentionally selected such a project as usability activities are many times taken care of in OSS projects by companies involved in the projects [1, 4, 5, 46], while OSS projects without company involvement usually are in need of outside usability support. We decided to focus our research efforts on such projects. In this case project there were no project leaders to be convinced that usability is important in order to gain access to the project and start doing usability activities for the software, as is typically the case in traditional software development. In fact, the project offered a fertile ground to enter into to begin with, as the importance of a good user interface was raised as a major design goal in the manual of the game, where goals for the user interface were described in the following way: the game is to have a ‘painless’ interface that supports game play and has support for ‘newbies’. Furthermore, the interface is to be designed so that game play becomes ‘easy’ and not ‘tedious’. (Project documentation)

The Student Usability Teams

Access to this case was gained through four student usability teams, UKKOSS 5, UKKOSS 6, UKKOSS 7 and UKKOSS 8 doing project work, which were aimed at introducing usability activities into OSS development project under close supervision of experienced HCI researchers. The students conducting the usability activities had usability background from at least two previous usability courses and software development background from at least three software development courses. Each student usability team consisted of four to six students working between 100 to 200 hours each in planning the usability activities, carrying out these planned usability activities, contributing to the development through code patches based on the results from the usability activities, communicating with the OSS project about these results, following up the impact of usability activities, collecting data and writing a project report.

The major intervention related to the experimentation with the participative approach was carried out during the UKKOSS 5 student usability team’s work. The other student usability teams have provided additional material related to the case OSS development project and related to the impact of the conducted usability activities as well as some insights related to the use of participative approach or lack thereof. One of the developers of the OSS project was a member of the UKKOSS 5 team and acted as both insider informant for all of the student teams and as a usability champion within the OSS project community marketing the results of the UKKOSS teams and trying to generate interest in usability and in the suggested changes.

The student usability teams had to first familiarize themselves with the game and technology in hand. This was done by the students individually so that everyone could get to know the technology as well as possible and to make up their own opinions of the technology and usability without group pressure. For usability tests, the student usability teams tried to find the best possible users based on the requirements received from the developers. The usability tests were done in controlled test environment in a usability testing laboratory where the student usability team could observe the users and their actions. Observations were done with multiple cameras and microphones and the video and

audio of each test session was recorded for further analysis. The users were supported by a team member if the users really needed help in order to continue the test. During the usability testing the influence and involvement of the usability team was kept as minimal as possible so that the users had to rely on the user interface and the software help features as much as possible. Video recordings from the usability test sessions were transcribed and analyzed. The student usability teams, in addition to identifying usability problems, also created suggestions for fixing the problems. (In line with the guidelines e.g. in [12, 42]). In addition to usability tests in the laboratory environment, the student usability teams carried out heuristic evaluations and cognitive walkthrough, used questionnaires in connection with the usability tests and interviewed the test persons after the usability testing sessions. Furthermore, the student usability teams carried out some development work and submitted UI improvement suggestions, level designs and code patches. Next more details related to the work done by each student usability team will be presented as well as some findings related to their work will be described.

The Work of UKKOSS 5 Student Usability Team

The UKKOSS 5 team consisted of five students who conducted comprehensive usability testing with six test participants and further analyzed the results from the tests and also from post-test interviews and questionnaires. Based on these usability tests the team found 44 different usability problems. The team also did heuristic evaluation for the game using game usability heuristics by Pinelle et al [36]. The team found 30 different usability problems through heuristic evaluation. Overall the UKKOSS 5 team found 62 different usability problems and reported these to the OSS developers and community by sending first a preliminary usability report and later a full usability report to the OSS project.

The preliminary usability report, delivered as an e-mail to the project's mailing list, resulted in active discussion on the email list with 53 emails related to it. The final usability report was delivered to the wiki of the OSS project. The developers commented actively on the usability issues and the suggestions to fix them. One of the developers added a comment field to each of the descriptions of the usability problems in the wiki, facilitating and encouraging discussion of the problems. This was something that the UKKOSS 5 team did not think of when adding the usability report to the OSS project wiki. Two of the developers added their comments to the reported usability problems, reporting that they had done something to this particular problem or discussed solutions to resolve this problem. Also two non-developers added comments. The student usability team members also replied to the developer comments, opening a dialogue between the developers and the usability team members. There have been developer comments to keep track of progress of the issues even one year after the usability report had been added, long after the UKKOSS 5 student usability project itself had ended. The reception of the usability reports within the developers and community was overall very positive and enthusiastic, though there were also critical opinions voiced against some of the suggestions in the reports.

In addition to usability testing and reporting, the UKKOSS 5 team submitted code patches and level design work, including new menus and a new tutorial for the game. These were received positively, and they were accepted into the code repository. The development team member who participated in the UKKOSS 5 did not commit the patches made during the project to ensure that the other development members considered them useful. Another team member from the UKKOSS 5 team was invited to the development team and given commit rights as a result of his contributions, for his active participation in discussions and for his recognized skills as a player. This can also be seen as a mark of a successful strategy for a usability expert to get into OSS project and gain recognition and merit.

Very important sign of success for any usability expert is the ability of her work to have an impact on the actual source code. Related to this, we took a look at the commit messages of this OSS project. The UKKOSS 5 team, their project work or report was directly referenced in commit messages four times. Reflecting the participatory nature of the team's work, one of these commit messages asked for input from the UKKOSS 5 team after making some changes to the tutorial authored by the UKKOSS5 team. This also illustrates that the student team's input on usability questions was valued by the developers.

There were also indirect references and other activity inspired by the student usability team's work. After UKKOSS 5's tutorial and other code patches were committed to the project's repository, several of the developers committed further tweaks and new features to build upon the introduced changes. Even during the development of the tutorial, a development team member, in communication via the project's IRC channel, provided new features for level design on short notice, so that the student usability team could implement the tutorial as planned. There were also commits that addressed issues discussed in the student usability team's reports, even though the reports or the team weren't directly referenced.

The Work of UKKOSS 6, 7 and 8 Student Usability Teams

The UKKOSS 6 and 7 student usability teams continued on the footsteps of the UKKOSS 5 usability team, but they had less resources to spend on usability activities, therefore their work relied less on the participative approach while it still offered us interesting information of this case OSS development project and of the long term impact of the conducted usability activities in the sense of changes in both the user interface and in the developer attitudes.

The UKKOSS 6 team consisting of six students and UKKOSS 7 team consisting of four students also carried out comprehensive usability tests and further analyzed the results from the tests, post-test interviews and questionnaires. The user interface of the game was also evaluated through cognitive walkthrough and compared to a set of usability heuristics and game usability heuristics by Pinelle et al [36] and Desurvire et al [11] in order to identify apparent problems in usability. Selection of the user test participants was done by asking possible users their gaming habits and past experience so that their experience and background would match the requirements from the development team – most of the testers in the usability tests carried out by the UKKOSS 5 team had had previous experience in the roguelike genre, and this time the development team wanted gamers without such experience. The UKKOSS 6 and 7 teams each found three different kinds of test participants to help them to test the game from three different perspectives of different kinds of users. The UKKOSS 6 team evaluated the game as a whole while the UKKOSS 7 team focused on evaluating the tutorial of the game.

The usability testing teams found several problems with the user interface and usability of the game. UKKOSS 6 team found 28 usability problems through heuristic evaluation and 11 usability problems in usability testing. UKKOSS 7 team found 10 usability problems through cognitive walkthrough and 16 usability problems in usability testing. Many of these problems were classified as severe problems that should be addressed as soon as possible. The user experience of the users participating in the usability tests ranged from gaining a possible new user for this game to a user that did not ever want to see the game again. The results of these tests were reported to the developers and the community and were also added to the usability project wiki page through the contact person.

There have not been comments by the developers to the usability problems reported by the UKKOSS 6 and UKKOSS 7 teams on the wiki page; however, one of the non-developer contributors of the OSS project converted the usability report from its original pdf form into proper wiki page. It can be argued that also this kind of initiative from within the community shows that the usability activities and their results were considered important to the community. Also, the developers acknowledged the good quality of the reports, indicated that they have read them, praised the usability activities as being professional and maintained that the issues raised in the reports should be fixed. On the other hand, the UKKOSS 6 and 7 student usability teams were not as much actively participating within the community as the UKKOSS 5 team and the communication between the teams and the developers and the community was mostly done through the developer usability champion who was a member of the UKKOSS 5 team.

The work of the UKKOSS 8 student usability team has only recently concluded, and its impact on the case project is still not fully clear. The team was similar in resources to UKKOSS5: that is, in addition to testing and analyzing, it also had resources to provide coding and level design effort. The approach of this team was to not to participate in the case OSS development project in the way UKKOSS5 did, but to work with the usability champion. The team worked on the tutorial of the game based on the usability findings and suggestions of the previous UKKOSS teams. They found 29 different usability problems in their five usability tests and used the results to improve the tutorial levels. The scope and target of testing and development was agreed upon with the usability champion at the beginning of the project, and the results were handed to him at the end.

Certain problems were identified with this approach. Relying on the usability champion, instead of direct communication with the development team and community, to deliver the results to the case project resulted in a delay measured in months in this case. There was a miscommunication that made the student team mistakenly think the user interface was to be overhauled by the development team in the near future, which caused them to not include problems directly related to the user interface into the scope of their work. Additionally, the usability champion did not consider the team's output as immediately acceptable for the game, but has been reworking it for inclusion. This has also caused a long delay in the team's work finding its way into the game.

Generally, there has been no substantial direct commentary from the development team and the community on the UKKOSS8's work, apart from the usability champion. However, the improvements to the tutorial are included in the release schedule, which can be seen as an approval from the development team. Also, there have been improvements to the technical framework of the tutorial message system, likely inspired by the other changes to the tutorial happening at the same time.

One additional noteworthy observation related to this OSS project and to the position of usability activities in it emerged during the work of this team: some of the previously more usability oriented core developers who started to encourage the usability teams, wanted to have more usability activities and developed the proposed changes to the user interface have now been less active within the community. This might have an impact on the success of this team's work, although it is not a necessity to have those usability oriented core developers involved to gain a positive outcome.

DISCUSSION

Based on the results of this case study, we claim that the participative approach we selected proved to be successful and invaluable in introducing usability activities into the OSS development project. There already have been studies

where usability student teams have introduced usability activities into OSS development projects but there has not previously been this kind of a long-term research intervention involving one OSS case project and multiple usability student teams spanning multiple years. Even though the participative approach for introducing usability activities has also earlier been experimented with, in this case the student usability team was even more active within the community: they tried to make themselves known by participating in discussions, reporting game bugs, playing on public servers and editing wiki pages, as well as through submitting code patches, their work in this case resulting in having an impact on the OSS game. Therefore, the case shows that not only usability activities were carried out, but those actually impacted the solution, which is not necessarily the case regarding usability activities - neither in traditional commercial software development nor in OSS development (see e.g. [1, 22]). We claim that it is important that the usability activities show in the commit feed of the project, because all developers read commit messages. Therefore, it would be particularly important for new usability teams without an insider usability champion to submit patches, get them committed through developers with commit rights and this way make their presence visible and know among the developers and the whole community.

As mentioned, one of the members of the UKKOSS5 team gained commit rights and a status as a developer because he was participating actively in discussions and recognized as a skilful and committed player of the game. However, we do not claim that it is necessary that all usability team members have skills in implementing improvement patches or interest in gaming. The usability team members may not have to like gaming in general or the particular genre of games, but they would still have to play the game enough in order to know the game mechanics. In addition, the activities of this team member clearly enabled that person to build trust and show merits that are issues considered highly important related to introducing usability activities into OSS development [1, 4, 5]. If not being able to code or not interested at all to play the game any more than absolutely necessary, the usability expert can still try to be as active as possible in the discussions within the community, as it has been argued that power in OSS projects is not only material and technical, but also discursive, and one can gain authority not only through controlling the source code and by possessing technical knowledge and skills in programming, but also through controlling the discussions in the discussion space [43]. If usability experts become visible through active discussions, they may also be able to gain more authority and their work better impact.

Although the work of the UKKOSS 6 and 7 usability teams and the importance of fixing the usability problems they had identified were recognized by the core developers, there has not been as much practical impact as in UKKOSS 5. One might suggest that the UKKOSS 5 usability team would not have had so much chance for success if the team members had not participated actively within the community, submitted patches and been interested in gaming in general and playing that game in particular (or more generally in certain product domain and the particular product in question). The different approach taken by the UKKOSS8 team, relying on the usability champion instead of working directly with the development team and community, resulted in significant overhead in time and developer effort. If the process had been more iterative, the student team would have gotten more feedback from the developers and improved their work before submitting the final version. Additionally, miscommunications during the start of the project would likely have been cleared. Although the student team did contribute patches, compared to the iterative and participative approach of the UKKOSS5 team, the work of the UKKOSS8 team is taking much longer to get into the game's codebase, and there have been fewer developers taking part. These observations indicate, altogether, that there might be a connection between the level of participation and the practical impact of the usability work as seen in changes in the user interface and in improved usability.

The long-term case study enables observing the case OSS project and the process of introducing usability activities into the project during multiple years. During these years there have been noticeable changes within community and its culture. The ways how the developers and the community communicate have changed within the case OSS development project since the involvement of the first student usability team. The previously used mailing list is no longer as active as before, and the IRC channel for developers has emerged as the main mode of communication. Furthermore, some of the core developers who started championing the usability activities have now been less active within the case community, as mentioned. Although there have not been references to the UKKOSS team reports in the commit messages recently, there has been references to usability and user interface improvements. Nevertheless, the natural and gradual change in OSS communities presents a challenge for introducing usability activities into OSS projects because change within development team means that the usability team may have to constantly be active as new developers may not be familiar with usability and previously conducted usability activities. Altogether, usability experts as well as OSS researchers should take into account that the culture and the ways of working in an OSS project are not fixed but may change over time when developers, community members and their opinions change.

In this case OSS project, clear motivations could be identified for the usability activities. The importance of good user interface was identified as a major development goal in the game manual and it was seen important to support new users in particular. Therefore, usability was identified as a possible way to provide a competitive edge

for this OSS project, i.e. through attracting new users through superior usability. In fact, reviews and commentaries of this particular game often highlight it's relatively highly developed interface, and recommend it for beginners in the genre. In this OSS project, the benefits of better usability were identified but not actively discussed or used as primary motivators for the usability activities, as has been suggested by researchers dealing with usability cost benefit considerations.

Regarding motivations for usability in the game context in general, one can argue that usability may contribute to higher user satisfaction and productivity, even though in a slightly different meaning compared to applications supporting work tasks while game playing is voluntary [25]. In addition, OSS projects may benefit of usability work through reduced need for changes due to bad usability and through increased popularity of their software (see also [39]). On the other hand, as it has been argued that usability needs to be adapted to fit the game context (e.g. [25]) as well as to fit the OSS development context [39]. Therefore, the complex relationships between games, usability and open source software should be explored further.

CONCLUSION

This paper inquired the introduction of usability activities into OSS development through an empirical case study in the OSS development context. Our empirical analysis suggested that it is beneficial to introduce usability activities into OSS development project through a participative approach where the usability experts become recognized part of the development community through adapting their ways of work into the culture of the OSS project and submitting code patches. This kind of participative approach had a clear impact in the case project and usability activities were therefore successful.

This study provides several practical implications. First of all, usability experts interested in contributing to OSS projects may utilize the findings of this study and try the participative approach when entering the project. Furthermore, usability experts working in the company context may also consider how to apply these findings in their work. No doubt the participative approach is valuable also in the commercial context, but naturally the guidelines presented in this paper are all not directly applicable. For instance, submitting code patches may not be a culturally compatible way to enter the project, and active participation in discussions and interest in the use of the product in question may not yield desirable results. In the company context it needs to be considered anew what it entails to become a fellow member in the development and to align the ways of work with the engineers.

Regarding the limitations of this study, one must mention that it might be considered as a limitation that the usability experts in this case were students. However, this might not be such a problem as in OSS development the formal status (e.g. educational degree) of people, including the usability experts is not important but rather how people, i.e. the usability experts in this case, have contributed to the OSS project and community. Developers in OSS projects listen to rational reason and do not look at formal status. The core developers and community in general seemed to be satisfied when the usability teams – or any contributor for that matter – had contributed something for the common effort, no matter whether the usability teams consisted of students not yet experts in usability activities or consultant level usability experts. Actually, the developers thanked the good quality of the reports and even praised the work of the student usability teams as being of professional quality. In this case OSS development project all the UKKOSS teams were very open that they were students and doing the usability activities as part of their course work, and the developers and the community in general did not see their status as students as being any kind of problem.

There are also other limitations concerning the results. The results are based on the analysis of only one, and naturally very specific, case OSS development project that happened to be involved in game development and that was not commercially supported one. It is unclear how the nature of the product under development affected the results – i.e. whether the fact that the project was developing a game affected the success of the participative approach. On the other hand, the fact that there was no company involved in this OSS development project was our choice. In case there had been a company taking part, the participative approach might have been equally applicable, but on the other hand, if the part taking company had taken the responsibility of usability work for the OSS solution, there probably would not have been that much interest in the external usability support in the project and the integration of the work of these external usability experts with those hired by the company would, altogether, have represented a completely new, although also very interesting, research topic.

Regarding paths for future work, we maintain that still more work needs to be done related to inquiring how to introduce usability activities into OSS development, addressing different kinds of OSS projects (e.g. large, small, with different leadership styles, the OSS solutions representing different product domains, etc.) After introduction, usability also needs to be institutionalized (see [45]), and this needs future research in the OSS context, too. In addition, the complexities involved with improving usability in the open source game context need to be explored further. We also wish to emphasize that usability activities and methods originate from academia or from commercial software development industry, which have clear cultural differences to OSS development. Therefore, research on the appropriate and culturally compatible ways for combining usability and OSS development philosophies and practices is still needed.

REFERENCES

1. Andreasen, M., Nielsen, H., Schrøder, S. and Stage, J. Usability in Open Source Software Development: Opinions and Practice. *Information Technology and Control 25,* 3A (2006), 303-312.
2. Allen, C.D. Succeeding as a Clandestine Change Agent. *Communications of the ACM 38*, 5 (1995), 81-86.
3. Aucella, A. Ensuring Success with Usability Engineering. *Interactions* May + June, (1997), 19-22.
4. Bach, P. and Carroll, J. FLOSS UX Design: An Analysis of User Experience Design in Firefox and OpenOffice.org. *In Proc. OSS 2009* (2009), 237-250.
5. Bach, P., DeLine, R. and Carroll, J. Designers Wanted: Participation and the User Experience in Open Source Software Development. In *Proc. CHI 2009* (2009), 985-994.
6. Bevan N. *Cost Benefit Analysis*. TRUMP report, September, 2000.
7. Bloomer, S., and Croft, R. Pitching Usability to Your Organization. *Interactions* Nov.+ Dec. (1997) 18-26.
8. Bødker, M., Nielsen, L. and Orngreen, R. Enabling User-Centered Design Processes in Open Source Communities. In N. Aykin (ed.), *Proc. Human Computer Interaction International, Part I: Usability and Internationalization.* LNCS 4559 (2007), 10-18.
9. Cetin, G., Verzulli, D. and Frings, S. An Analysis of Involvement of HCI Experts in Distributed Software Development: Practical Issues. *In Proc. Human Computer Interaction International: Online Communities and Social Computing*, D. Schuler (ed.), LNCS 4564, (2007), 32-40.
10. Cooper, A., and Reimann, R. *About face 2.0: the essentials of interaction design.* Wiley, Indianapolis, 2003.
11. Desurvire, H, Caplan, M. and Toth, J. Using Heuristics to Evaluate the Playability of Games. *In Proc. CHI 2004* (2004).
12. Dumas J. and Redish J. *A Practical Guide to Usability Testing.* Intellect Books, Oregon, 1999.
13. Ehrlich, K., and Rohn, J. Cost Justification of Usability Engineering: A Vendor's Perspective. *In Cost-Justifying Usability*, Bias, R. and D. Mayhew (eds.), Academic Press, 1994, 73-110.
14. Gill T.G. and Hevner, A.R. A fitness-utility model for design science research. *In Proc, DESRIST 2011,* LNCS, 6629(2011) 237-252.
15. Giuri, P., Ploner, M. Rullani, F. and Torrisi, S. Skills and Openness of OSS Projects: Implications for Performance (Working Paper). Laboratory of Economics and Management, Sant'Anna School of Advanced Studies, (2004) Pisa, Italy.
16. Grudin, J. Interactive Systems: Bridging the Gaps between Developers and Users. *IEEE Computer 24,* 4 (1991), 59-69.
17. Gulliksen, J., Boivie, I., Goransson, B. Usability professionals—current practices and future development. *Interacting with Computers 18* (2006) 568-600.
18. Hevner, A.R. The Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems 19*, 2, (2007), 87-92.
19. Hevner, A.R., March, S.T. and Park, J. Design research in information systems research. *MIS Quarterly 28*, 1 (2004), 75-105.
20. von Hippel, E. and von Krogh, G. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science 14*, 2 (2003), 209-223.
21. Hutchings, AF., and Knox, S.T. Creating Products – Customer Demand. *Communications of the ACM 38*, 5 (1995), 72-80.
22. Iivari, N. Understanding the Work of an HCI Practitioner. *In Proc. 4th Nordic Conference on Human Computer Interaction,* A. Morch, K. Morgan, T. Bratteig, G. Ghosh, D. Svanaes (Eds.) (2006) 185-194.
23. ISO 13407. Human-centered design processes for interactive systems. *International standard* (1999).
24. ISO/IEC 9126-1. Software Engineering, Product quality, Part 1: Quality model. *International Standard* (2001).
25. Jorgensen, A.H. Marrying HCI/Usability and Computer Games: A Preliminary Look. *In proc. NordiCHI 2004*. (2004).
26. Karat, C.M. A Business Case Approach to Usability Cost Justification. *In Cost-Justifying Usability*, R. Bias, D. Mayhew, (eds.), Academic Press (1994) 45-70.
27. Ljungberg, J. Open Source Movements as a Model for Organizing. *European Journal of Information Systems 9*, 4 (2000) 208-216.
28. Mayhew, D. Strategic Development of Usability Engineering Function. *Interactions 6*, 5 (1999) 27-34.
29. Mayhew, D. *The usability engineering lifecycle: a practitioner's handbook for user interface design*. Morgan Kaufmann Publishers, San Francisco. 1999.
30. Mayhew, D. and Mantei, M. A Basic Framework for Cost-Justifying Usability Engineering. In *Cost-Justifying Usability*, Bias, R., and Mayhew, D. (eds.) Academic Press (1994) 9-43.
31. Nichols, D. and Twidale, M. The Usability of Open Source Software. *First Monday 8*, 1 (2003)
32. Nichols, D. and Twidale, M. Usability Processes in Open Source Projects. *Software Process Improvement and Practice 11* (2006) 149-162.

33. Nielsen, J. *Usability engineering*. Academic Press, Boston, 1993.

34. Ohnemus, K. Incorporating Human Factors in the System Development Life Cycle: Marketing and Management Approaches. *In Proc IEEE IPCC96*, (1996) 46-53.

35. Peffers, K., Tuunanen, T., Rothenberger M. and Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems 24*, 3 (2007) 45-77.

36. Pinelle, D., Wong, N. and Stach T. Heuristic Evaluation for Games: Usability Principles for Video Game Design. *In Proc. CHI 2008*. (2008)

37. Rajanen, M. Different Approaches to Usability Cost-Benefit Analysis. *In Proc. ECITE 2006*, Remenyi, D. and A. Brown (eds.), (2006) 391-397.

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

39. Rajanen, M. and Iivari N. Traditional Usability Costs and Benefits – Fitting them into Open Source Software Development," *In Proc. 18th European Conference on Information Systems (ECIS 2010)*, (2010)

40. Rajanen, M., Iivari, N. and Anttila K.: Introducing Usability Activities into Open Source Software Development Projects – Searching for a Suitable Approach. *Journal of Information Technology Theory and Application (JITTA) 12*, 4 (2011), 5-26.

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

42. Rubin, J. *Handbook of Usability Testing: How to plan, design, and conduct effective tests.* John Wiley and Sons. New York. 1994.

43. Sack, W., Détienne, F., Ducheneaut, N., Burkhardt, J., Mahendran, D. and Barcellini, F. A methodological framework for socio-cognitive analyses of collaborative design of open source software. *Computer Supported Cooperative Work 15*, 2 (2006), 229–250.

44. Scacchi, W. Understanding the requirements for developing open source software systems. *IEE Proceedings – Software 149*, 1 (2002), 24-39.

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

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

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

48. Zhao, L. and Deek, F. Exploratory inspection: a learning model for improving open source software usability. *In Extended Abstracts of the CHI2006* (2006) 1589-1594.