1-1-2010

Traditional Usability Costs and Benefits: Fitting them into Open Source Software Development

Mikko Rajanen
University of Oulu, mikko.rajanen@oulu.fi

Netta Iivari
University of Oulu, netta.iivari@oulu.fi

TRADITIONAL USABILITY COSTS AND BENEFITS – FITTING THEM INTO OPEN SOURCE SOFTWARE DEVELOPMENT

TRADITIONAL USABILITY COSTS AND BENEFITS – FITTING THEM INTO OPEN SOURCE SOFTWARE DEVELOPMENT

Rajanen, Mikko, Department of Information Processing Science, University of Oulu, P.O Box 3000, 90014 Oulu, Finland, mikko.rajanen@oulu.fi

Iivari, Netta, Department of Information Processing Science, University of Oulu, P.O Box 3000, 90014 Oulu, Finland, netta.iivari@oulu.fi

Abstract

Usability is an important quality characteristic of software (SW) products and systems. Usability cost-benefit analysis models outline the potential benefits and costs of usability. These models are important for selling the usability activities for management. The implications of open source software (OSS) development context on usability cost-benefit considerations have not been addressed. Based on the insights of literature addressing the OSS development context, as well as two interpretive case studies carried in the OSS development context, the paper reveals that there are interesting parallels and differences when considering the costs and benefits of usability in OSS development context. Based on the analysis, a refined model of usability costs and benefits fitting the OSS development context is introduced. The model can be used for selling the usability activities to OSS development. Implications for theory and practice are discussed.

Keywords: Usability Cost-Benefit Analysis, Open Source Software Development

1 INTRODUCTION

This paper examines usability costs and benefits in open source software (OSS) development. The status of usability in OSS development today resembles the status of usability in traditional software development about twenty years ago with little or no usability specialists and no usability activities integrated to the development process. Therefore, we assume that the measures used for selling usability to traditional software development could now be of use in OSS development, but with some modifications.

A common definition of usability is: "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 13407). Usability is an important competitive edge in maturing software markets (Grudin 1991, Nielsen 1993, Rosson and Carroll 2002). However, bringing usability into the software development life cycle is a challenge (Ohnemus 1996, Iivari 2006, Rajanen & Iivari 2007). One reason for this is that the benefits of usability are not easily identified or calculated. Usability has been competing for resources against other project groups, who have objective cost-benefit data available (Karat 1994).

Usability cost-benefit analysis models have been introduced to tackle the problem of estimating and revealing the costs and benefits of better usability (see Donahue 2001, Bevan 2000, Ehrlich & Rohn 1994, Karat 1994, Mayhew & Mantei 1994). These usability cost-benefit analysis models are to be used to persuade and manipulate management and developers to introduce usability considerations into the development process. There is some research analyzing the existing usability cost-benefit analysis literature and the potential risks of using usability cost-benefit approach to promote usability in company software development context (Rajanen et al 2007, Rajanen 2007, Rajanen 2006, Rajanen 2003). However, in this paper these models are contrasted with the OSS development context, which is a current development context constantly growing in importance (Fitzgerald 2006, Niederman el al. 2006). There is a lack of research on how usability as well as its cost-benefit analysis fit into this new, highly influential development context. This paper answers the following research question: "*How should the potential usability costs and benefits from traditional usability cost-benefit analysis literature be refined to fit them into the open source software development context?*"

OSS means software of which source code is available for everyone to read and modify. The fundamental idea is to enable software to evolve by exploiting community participation. OSS makes possible also for end-users to adapt software to their personal needs and fix defects (Raymond 1999). OSS community is often depicted with an onion model. Project leaders making decisions in the project form the hard core of the onion. They are supported by committers having direct write access to the project's source code, but required to ask permission for major modifications before committing a change. Contributors are external developers and users, who send bug reports and minor fixes for bugs. Although they can download and modify the source code, they do not have the power to upload their modifications into the project's official source code repository. The outmost layer consists of end-users who do not participate in the community, but only use the software (Aberdour 2007).

Currently, also companies have started to exploit OSS in their business. Utilizing OSS applications and development tools has been common for a long time, but also the utilization of the actual source code is becoming popular. The availability of free, ready-made components can reduce the development costs substantially. Lately, companies have also started to participate in OSS communities and even to launch new communities for their products. (Dahlander & Magnusson 2005, Fitzgerald 2006, Iivari et al. 2008, Niederman el al. 2006). The revenue models of OSS have been changed from pure support selling and loss-leading to more comprehensive marketing and sales management and servicing eund implementation (Rajala et al. 2001).

In this paper the focus is both on the traditional community OSS development, which is organized according to the onion model outlined earlier, and on the company OSS development (Iivari et al 2008), in which commercial software development builds application software on top of OSS, and

potentially also releases the source code for the OSS community to develop it further. Two interpretive case studies (cf. e.g. Klein and Myers 1999) addressing these two OSS development models have been carried out. The results of the case studies will be combined with the literature on OSS development and their implication on the usability cost benefit models is examined. In both cases the focus is on OSS solutions that are targeted at a mass of users, who do not necessarily have IT education or programming skills. Traditionally, technically very skilled developers have developed OSS for their own use, but nowadays OSS solutions have more and more users without deep technical knowledge. Therefore, usability of OSS has become very important, but currently, however, it tends to be quite poor (Andreasen et al. 2006, Benson et al. 2004, Cetin et al. 2007, Nichols & Twidale 2003, Nichols & Twidale 2006, Zhao & Deek 2005, Zhao & Deek 2006).

The paper is organized as follows. The next section reviews literature on usability cost-benefit analysis. The third section presents empirical insights based on case studies carried out in the OSS development context. The fourth section introduces a refined model of usability costs and benefits fitting open source software development context. The final section discloses the central themes and observations of the paper, outlines the limitations of this study, and suggests paths for future work.

2   USABILITY COST-BENEFIT ANALYSIS MODELS

There are not many published models for analyzing the costs and benefits of usability in software development. Most of the usability cost-benefit models presented in this paper (i.e. Mayhew et al 1994, Ehrlich et al 1994 and Karat 1994) were selected from the book Cost-Justifying Usability, by Bias & Mayhew (1994). The analyzed models were selected, because they represent the variety of different views for usability cost-benefit analysis. In addition, Donahue's (2001) and Bevan's (2000) usability cost-benefit models were included, because they represent slightly different views to business benefits of usability. The approaches for usability cost-benefit analysis and the potential benefits and costs of better usability presented in the current usability cost-benefit analysis literature do not take the OSS context into account.

The usability cost-benefit literature has many approaches to usability cost-benefit analysis and where the analysis should be focused on. The focus can be on the benefits that are of most interest to the audience of the analysis (Mayhew et al 1994). The focus can be split between costs for the development organization and benefits for the customer organization (Donahue 2001). The approach for usability cost-benefit analysis can be to analyze the potential benefits of better usability to the organization during development, sales, use and support phases (Bevan 2000). The focus can be on the potential benefits of better usability from the viewpoint of vendor company, corporate customer and end user (Ehrlich et al 1994). The focus can be on analyzing the usability benefits through cost-benefit calculation of human factors work (Karat 1994).

We cover the different approaches to usability cost-benefit analysis by combining the identified usability costs and benefits from the different usability cost-benefit analysis models based on previous research (Rajanen et al 2007, Rajanen 2007, Rajanen 2006, Rajanen 2003). We believe these usability costs and benefits can be refined to fit into OSS context by taking into account the special characteristics of the OSS development.

The usability cost-benefit literature has many approaches to identify the potential benefits of better usability in development context and use context. The potential benefits of better usability in development context are categorized as increased sales, reduced development costs and reduced training and support costs. The potential benefits of better usability in use context are categorized as reduced training time, increased productivity and increased customer and user satisfaction. (Donahue 2001, Bevan 2000, Mayhew et al 1994, Ehrlich et al 1994, Karat 1994).

The usability cost-benefit literature categorizes the costs of better usability into one-time costs, recurring costs and redesign costs. One-time costs are costs from establishing the infrastructure for usability work such as usability test laboratory. The recurring costs of usability are identified as

continuous costs of usability work such as salaries of usability specialists. The prototype related redesign costs are additional costs related to each of prototype iterations. In addition to official training, there are also hidden costs of peer-support when end users seek help from their expert colleagues. (Donahue 2001, Bevan 2000, Mayhew et al 1994, Ehrlich et al 1994, Karat 1994). Table 1 summarizes the costs and benefits of usability outlined by the usability cost-benefit analysis models.

	Development context	Use context
Usability benefits	Increased sales Reduced development costs Reduced training and support costs	Reduced training time Increased productivity Increased (customer and user) satisfaction
Usability costs	One-time costs Recurring costs Redesign costs	-

Table 1.   *Summarization of the usability costs and benefits outlined in models*

3   CHARACTERIZING USABILITY IN OSS DEVELOPMENT

The results of two case studies are discussed in this paper. The first case has been discussed in (Iivari 2008, Iivari 2009) and the second case in (Iivari et al. 2008). The cases were originally examined from the viewpoint of usability and user participation in OSS development, but they were revisited from the viewpoint of usability costs and benefits. The cases and the research methods utilized related to them are next briefly outlined. More information is found in (Iivari 2008, Iivari 2009, Iivari et al. 2008).

The first case is a traditional community based OSS development project developing a media application. The project has shown clear interest in improving the usability of their solution. They are, e.g., listed in a website requesting usability support from usability specialists to OSS projects. In addition, a usability discussion forum has been established, encouraging the users to suggest improvements to the OSS. There are 9 developers with committer rights in this project (sourceforge.net). They will be called 'developers' in this paper, the rest of the people being called 'users'. The case represents a 'traditional', small OSS project without corporate resources that initially tries to deal with usability. It is already known that in large projects or in projects, where also companies are involved, there might be usability resources available (see Benson et al 2004, Frishberg et al 2002, Iivari et al 2008, Nichols et al 2006). The interpretive case study described here relies on the data gathered from the project's website, particularly from the usability discussion forum. The forum included altogether around 1600 messages and 400 topics. The forum is the place from which the role of usability in the project can be revealed, in anywhere.

The second case, on the other hand, provides data related to the company involvement in OSS development. From the viewpoint of usability, important is that there are usability specialists in the case taking care of usability in the development. The case is a software development unit of a large, global corporation. The unit has been responsible for taking part in OSS development in the company. This has been started already few years ago. The unit also has a strong background in usability and user interface (UI) development. This case offers a rich setting to analyze the emerging company OSS development phenomenon with strong emphasis on usability. The interpretive case study described here relies on the data gathered from the unit and associated OSS projects. In the unit, two people with the job title developer, two with that of usability specialist and one with the title of manager were interviewed. In addition, two managers participated in meetings, in which issues related to usability and user participation were discussed. Memos of the meetings are also included as research material. In addition, the data includes blogs kept by the usability specialists, the company's OSS related websites and the associated OSS projects' websites.

The data gathered from both cases has originally been analyzed from the viewpoint of usability and user participation, but now it is revisited to check what it can reveal from the viewpoint of usability costs and benefits. In this analysis, all the data gathered was carefully read through and all relevant (according to table 1) pieces of information were extracted and categorized as 'benefits' or 'costs' of usability or as having implications on the 'benefits' or 'costs' of usability as outlined in the existing models. The analysis was data driven, the Table 1 however acting as a sensitizing device. The existing OSS literature was also used to supplement the empirical data derived from these two cases.

3.1   Community OSS Development

The interest this OSS project has in usability is evident. The usability forum begins with the invitation from one of the developers to users to participate in further improving the UI. In addition, the developers argue that 'happy users' make them 'happy'. The developers also highlight that users demand usability and are very 'critical on these issues'. Also the users emphasize that attracting new users with usability is important: *"A lot of non-technical users are changing to Linux and I think that the software should be ready for them. (…) There are three types of computer users in the world. The first one is definitely the majority and they don't understand how the software works and they don't want to. They just want it to work."* The users also highlight that a large user base helps the development as well as the UI work *"[The application] is the best [application] I currently know. If it gets more users, I think it will be merely good, as it will also bring more developers (and more users who cannot code, like me, but who are eager to help out with functionality/UI ideas)."* Therefore, happy users, attracting a larger amount of users and usability demanding users are the reasons for emphasizing usability in this case.

Four categories into which a large proportion of content of the messages fitted have been identified in the usability forum (Iivari 2009). The categories are: problems, feature requests, thanking, and helping. The helping messages offer advice for the sender, and due to these messages one can argue that the forum is used for describing how one can and is supposed to use the OSS, which has already been reported as the strength and user-centric aspect of OSS development: the OSS developers help the users in discussion forums and provide field support for the users (Lakhani & von Hippel 2003). Also in this OSS project, the users post questions, comments or suggestions, and more experienced people reply (helping, commenting) to them. In addition, in the thanking messages the senders thank the developers and/or praise the OSS as the best one available. One can argue that the thanking messages are also typical in the OSS forums, since the OSS developers are not paid for their work, but instead their own needs but also the reputation gained motivates them; a large user base motivates the developers and is a flattery and reward (Ye & Kishida 2003).

Despite the typicality of the previously mentioned messages in OSS forums, the bug and feature request messages can be argued of being the most characteristic to OSS forums. OSS development largely relies on the community model of development, where there is a large group contributing, some by developing the software, others by reporting bugs or proposing new features. In the feature request category, different kinds of features are requested. In the problem category, different kinds of problems are expressed. The sender does not know how to use a particular feature or the OSS altogether, or he/she does not like a particular feature or the OSS altogether. The effect of good usability on the users' feature requests is acknowledged in this project. A developer argues after a feature request that the user already can do that and argues that it is a problem if users ask for features that are already available.

Regarding developing usability in the analyzed project there are no usability specialists contributing to the project. Even though the project is listed in a website requesting support from usability specialists to OSS projects, no usability specialists have entered the project. The project does not seem to utilize any established usability methods either. There are no mentions of usability evaluations, user studies or user task redesigns carried out in the project. However, some message senders provide data related

to the 'usability reviews' they have carried out, the messages outlining usability problems identified from the OSS, without mentioning, however, the methods they have utilized in doing the review.

Nevertheless, the UIs are produced also in this OSS project. Altogether, there are 499 feature request messages and 300 problem messages counted in the usability discussion forum (Iivari 2009), indicating that UI redesign and evaluation is continuous and active in the project. There are UI designs in a form of mock-ups produced also by the users, related to which comments are asked and provided in the discussion forum: *"I've made a draft of what I mean: [a link to a mock up is provided]. (…) Well, what do you guys think? (You can vote for it [a link is provided])"* Sometimes, the developers also implement the suggestions very fast with a reply: *"Good idea, done ;)"* However, the developers might also respond simply that he does not like it and reject it. The users capable of coding can also make changes to the code: *"Hi! I miss two things in [the application]: [a feature] and [a feature]. I made a patch [a link is provided]"* However, again, it is up to the developers to make the decision whether to include these contributions into the code base. Altogether, the discussions in the discussion forum might lead to the changes in the OSS, but the developers have the power to decide about that.

3.2   Company OSS Development

In the company OSS development case analyzed in this paper OSS is mainly utilized as part of the company's products; OSS is used as a basis on top of which applications are developed, the top level remaining closed source: *"[The application is] closed source, on top of open source."* (Usability specialist) *"Openness did not spread to the UI code."* (Manager) However, in some situations the source code of the whole product has been released. Nevertheless, there are two reasons for keeping the UI closed source: UI solutions are perceived to be the competitive edge and legal checks that would be needed are considered as complicated and expensive: *"Usability and UI are so valuable that they are not released."* (Usability specialist) *"Sometimes there are difficulties due to the licenses. One has to know what kind of code is used, what libraries are used."* (Developer) *"[The company] has a lot of patents related to […]. There was no desire to carry out investigations."* (Manager)

In the case, usability and UI design are emphasized a lot: *"Everything goes in the end under the line. Of course [usability] affects many things, like brand, image. (…) We want that those people who buy our stuff will buy them again."* (Usability specialist) *"[The firm] makes devices for people. (…) Why would anybody buy a product (…) that costs as much as the competitors, but is 10 times more difficult to use? So maybe market is the answer [for usability]"* (Developer) There have also been investments in the company in usability and UI design: *"Usability and user experience are becoming (…) the key in today's market, the company started to invest a lot more in usability and user experience. (…) In the beginning, for a couple of years, there was just one guy taking care of everything of usability and user interaction and now there are seven."* (Usability specialist)

Related to the development of usability, the projects are carried out quite a similar way in the case. First, a user group for which the solution is developed is settled by the developers and usability specialists: *"Well, at first we sat down and thought that we are now making this kind [application] and thought that the display is this kind [display]. (…) So, this is the starting point and otherwise we have a lot of freedom to do what we want."* (Manager) Afterwards, users' task redesigns as well as UI design solutions are produced: *"We started by building use cases in a user centered way. We described the normal use situations that can be related to the software like this. After that we moved to producing rough UI. (…) Based on that, the first usability things were done, in which I was involved, evaluations."* (Usability specialist) Both expert evaluations relying on the expertise of the usability specialists and empirical user testing with real or potential users are carried out: *"First we produce rough UI (…) and afterwards (…) we carry out expert evaluation. In them we use one or more usability specialists and modify the UI according to them, and then move it to a more detailed level. After that we make a simulation of the UI and carry out a traditional usability test in a laboratory with a sufficient amount of users, from six to ten per iteration."* (Usability specialist) In the expert evaluations, a company tailored method is used: *"I have myself tailored [a method] to fit the needs of*

our team, it is like a combination of cognitive walkthrough and heuristic evaluation" (Usability specialist) Typically UI style guides are highly influential in this company, but they have been less influential in the company's OSS projects: *"In this case it was a starting point that we have a free hand to do the UI. It is in the [company's] environment quite an unusual situation. We did not have as a restriction any existing style, like UI style."* (Usability specialist)

During empirical evaluations, OSS communities are utilized to provide feedback in a fast and easy way: *"These open source software, they create a lot of communities, so it's a really, really rich place to gather user feedback."* (Usability specialist) The OSS community is seen as ready to use less than perfect early releases of the software and collaborate with the developers, and developers expected the community members to do so: *"We expected people contribute the UI"* (Usability specialist). *"Actually what we did was: hi, let's do a prototype in, for example, couple of weeks and see how it goes. The user interface is very, very, very simple. What we did was to make it and release it. (…) Involvement was all through emailing community: what do you like to see, what do you like to have? (…) We did use the community to find out more and to find out improvements."* (Usability specialist) Therefore, the continuous redesign and evaluation that has already been described related to community OSS development is initiated also the in company OSS development.

In the case, it is also emphasized that the usage of the OSS gives more time for usability, design and testing: *"We were able during much earlier phase to handle the finished software. (…) Usually it takes pretty long time to make a simulation of it. (…) Here we were able to experiment in practice what it looks like and how it feels like in real use. (…) This way it affected, positively."* (Usability specialist) *"The ability of doing late changes that of course the open source project provided us, because they give us ground work that is solid and mature. Then we can just build the experience on top. So actually in the product competition, we have time to do more testing or more designing and more late changes. So it indeed works and helps a lot."* (Usability specialist) Therefore, it is indicated in the case that the time saved in the implementation phase can now be spent to design, testing and redesign that is carried out together with the OSS community.

4   DISCUSSION

Based on the existing OSS literature and our case findings, we claim that the usability cost-benefit models need to be altered somewhat to fit the OSS development context. Before modifying them, however, we present our case findings related to the implications of the OSS on the existing usability cost-benefit analysis categories.

The increased sales was identifiable as a potential usability benefit category from both case studies, but in a somewhat different sense than traditionally. In community OSS development, the developers want to attract more users and keep their existing users happy. Large user base is an important source of status and peer recognition, even though there is nothing sold to the users like the usability cost-benefit models assume. In company OSS development, user satisfaction and good UI are important image and competitive factors for the company. These can have an impact to sales and they are in line with how usability cost-benefit models present this benefit (cf. Rajanen 2006, Rajanen et al 2007).

OSS has also certain implications on the category related to the reduced development costs. Early and continuous releases are characteristic to OSS development and they enable early feedback gathering and redesign. In community OSS development the community actively comments on and redesigns the solution during the project, possibly saving the developers' time and effort. Also in company OSS development the early releases give more time for designing and testing in cooperation with the community. This helps to identify the needs for redesign early in the project, when changes are still easy and cheap to do, which is in line with how usability cost-benefit models outline this benefit (cf. Rajanen 2006, Rajanen et al 2007).

Also implications for the category related to reduced training and support costs can be identified from OSS development. In OSS development, user support is provided as peer support in the OSS forums.

In the company case the OSS community might also provide the user support in the forums, which can be considered as a benefit, if the company is not interested in making money through providing it. Implications of the OSS on the existing usability benefit categories are summarized in Table 2.

Usability Benefits	Community Open Source	Company Open Source
Increased sales	More users. Happy users	Increased user satisfaction. Good UI as important image and competitive factor
Reduced development costs	Early releases give more time for redesign. Community takes active part in providing feedback and redesigning the solution in the forums	Early releases give more time for usability work and feedback, help to identify needs for redesign early. Community takes active part in providing feedback and redesigning the solution in the forums
Reduced training and support costs	Community provides peer support in the forums	Community provides peer support in the forums

Table 2.   Implications of the OSS on Usability benefit categories

OSS has also certain implications on the costs outlined in the models. One-time costs can naturally be identified from company OSS development where usability laboratory might be established and guidelines and tailored usability methods produced as usual. The existence of the OSS might also lead to the establishment of an infrastructure for the OSS community, which implies some costs. One-time cost of establishing a usability discussion forum can also be identified from community OSS development, even though this cost is quite marginal.

Recurring costs can be identified especially from company OSS development where usability specialists are hired, but this resembles the situation in traditional software development. On the other hand, there are no identifiable recurring costs in community OSS development, if there are no usability specialists or people carrying out their tasks in the project. However, redesign costs can easily be identified both in community and company OSS development. In company OSS development, more time and opportunities for redesign probably increases the redesign costs when compared to the closed source software development. In community OSS development, there is a lot of redesign going on all the time, because users and developers produce mock-ups and comment on them. The OSS community is more ready to use less than perfect early releases of the software and collaborate with the developers, and developers expect the community members to do so. Therefore, the redesign costs clearly can be associated with OSS development. Implications of the OSS on the existing usability cost-benefit categories are summarized in Table 3.

Usability Costs	Community Open Source	Company Open Source
One-time costs	Establishment of a usability forum	Establishment of a usability laboratory, guidelines, company tailored usability methods
Recurring costs	If usability specialists work in the projects, their time and effort is needed	Usability specialists hired as usual
Redesign costs	Uncontrollable continuous redesign. Potentially a lot of time spent in redesign. Users and developers produce and comment on mock-ups	Uncontrollable continuous redesign. Potentially increasing redesign costs since there is more time and opportunities for redesign than in closed source development. Users and developers produce and comment on mock-ups

Table 3.   Implications of the OSS on Usability cost categories

Next, a refined model of usability costs and benefits fitting the OSS development context is introduced. The model is to be used as a practical tool e.g. by usability specialists in selling usability activities to OSS development; to the core developers and to the whole community.

Regarding the benefits of usability, we argue that increased sales should be perceived rather as increased popularity or distribution of software. In the OSS context, users don't pay for the software, but large user base might be a source of reward and peer recognition for the developers. Potentially increasing the number of active and committed developer- and non-developer –users can be seen as a usability benefit in both community and company OSS development. In company OSS development some aspect of the product, like hardware, support, additional features or even the software, are sold to users. Therefore, increased popularity and distribution through better usability is in some way similar to what the usability cost-benefit models categorize as increased sales.

Reduced development costs can be achieved both in the community and company OSS development. Usability cost-benefits models approach the reduced development costs through less need for development resources and earlier time in market (Rajanen 2006, Rajanen et al 2007). In both community and company OSS development the community can take active part in redesign and therefore reduce the development time, even if the development project does not have any salaried positions and costs in the similar sense as is the case in the company context. The OSS development is very iterative and rapid by nature (Raymond 1999), so early releases to community give more time for iteration and redesign. Developers in both community and company OSS development can save time, when there is less pressure for redesign through better usability achieved already. Also the active dialogue with the community and the change requests in the forums can result in abandoning bad design solutions early in the project, when changes are easier to make. The change requests can be gathered more systematically and in early phase through usability activities. Also the non-technical users can contribute to the project in more manageable manner though usability methods.

As mentioned, company OSS development can use the community as a provider of training and support. If providing support is an important part of company's business model and sold as a service to customers and end-users, the community based peer support and better usability can potentially reduce company's earnings. But on the other hand poor usability can potentially harm the company and product image and decrease popularity, therefore reducing the support based earnings in any case (cf. Rajanen et al 2007). Community OSS development can have less need for peer support in forums through better usability, when non-developer users write less in community forums asking usability related questions or requesting features that already exist in software but are too difficult for users to find. Community based support, where users take an active part in helping each other, is seen as an important aspect in both community and company OSS development. This is clearly different to the categorization of this in the usability cost-benefit models, where peer-support is seen as a work slowing hidden cost in the use-context (cf. Ehrlich et al 1994, Rajanen 2006). The refined usability benefits fitting the OSS development context are summarized in Table 4.

Usability Benefits	Community Open Source	Company Open Source
Increased popularity and distribution	Increased number of active and committed developer-users and non-developer –users. Increased developer-user and non-developer-user satisfaction	Increased number of active and committed developer-users and non-developer –users (some of them also willing to pay). Increased developer-user and non-developer-user satisfaction
Reduced development costs	Less pressure for redesign through change requests in the forums. More systematic redesign	Less pressure for redesign through change requests in the forums. More systematic redesign
Reduced training and support costs	Less need for peer support in the forums	If company provides training and support, less need for them

Table 4.   Usability benefits in open source software development context

While the benefits of usability can be used as a motivator for introducing usability activities in OSS context, the costs of usability should be seen particularly in the community OSS development context as bringing realism and showing the usability specialists and developers what they need to be

committed to when introducing usability activities. As categorized in the usability cost-benefit models, the one-time costs for usability can be identified also in community OSS development in a sense that there needs to be an usability infrastructure, such as forums and usability guidelines for developers established, but no physical infrastructure is needed since OSS development is typically distributed. The company OSS development is similar to closed source development. A usability infrastructure, e.g. a usability laboratory, may already exist or it might need to be established, even though it is not mandatory. Company may also have one-time costs in establishing internet-based usability and OSS related infrastructures, such as forums or OSS repositories.

In company OSS development the recurring costs are very much similar to traditional closed source development where usability specialists are available and usually already working in the company. In community OSS development there seems to be a great need for usability specialists in OSS projects but the OSS projects have difficulties in attracting usability specialists to contribute (e.g. Benson et al. 2004, Cetin et al. 2007, Twidale et al 2005, Zhao et al 2005). There have been some efforts to attract usability specialists by listing OSS projects needing usability specialists in the web, but for some reason these efforts have not succeeded yet in bringing OSS projects and usability specialists together.

Redesign costs through usability activities are evident in both community and company OSS development where by the nature of OSS development there is uncontrollable and continuous redesign. This can potentially increase redesign costs in company OSS development and increase time spent on redesign in community OSS development. In both community and company OSS development the community can provide help by testing and redesigning but professional usability specialists are still needed in OSS projects. Communicating with the community about features, testing and redesign requires time in both community and company OSS development. New features are usually requested constantly in the community forums and because anyone can contribute to the project, this poses a new challenge to usability specialists contributing to community OSS projects, because the version they want to redesign may already be outdated before they even begin their work. The refined usability costs fitting the OSS development context are summarized in Table 5.

Usability Costs	Community Open Source	Company Open Source
One-time costs	Establishing internet-based usability infrastructure (usability forums, guidelines, methods, etc.), in distributed OSS development no establishment of physical infrastructure (e.g. a usability lab)	Similar to closed source development, physical usability infrastructure established. Establishing internet-based usability and OSS infrastructures (version control, bug reporting systems, discussion forums, etc.) for OSS communities
Recurring costs	Usability specialists time and effort, but difficulties in attracting usability specialists in OSS projects	Similar to closed source development, usability specialists salaries
Redesign costs	Time spent communicating with the community, potentially a lot of time spent on redesign. Community can help by taking part in testing and redesign but professional usability specialists needed	Time spent communicating with the community, potentially a lot of time spent on redesign. Community can help by taking part in testing and redesign but professional usability specialists needed

Table 5.   Usability costs in open source software development context

The existence of developer-users in OSS development context does not fit the categorization of the usability cost-benefit models, where the usability benefits are categorized separately for developers in the development context and for all users in the use context. In OSS development the benefits of usability for developer-user have an impact both in the development context and in the use context. The collaboration between OSS developers and users also opens new and interesting research opportunities and challenges for usability research, because the traditional distinctions between developers, usability specialists and users gets blurred in OSS development.

In addition, in OSS development the focus has traditionally been on technology and functionality, not on usability. OSS users have traditionally tolerated bad usability, if highly innovative technological solutions have been provided. For the non-developer users, however, usability may become a highly important factor, maybe even hindering the use of an OSS solution. If OSS projects want to attract this kind of users, they need to start to emphasize usability.

The introduction, best methods and possible benefits of usability in OSS context are relatively new areas for usability theory and practice. The importance of better usability was recognized both for closed source software development and for OSS development in our case studies. The usability research should introduce new and more suitable usability methods and practices for this emerging and important area of community and company OSS development.

5   CONCLUSION

This paper contrasted the existing usability cost-benefit analysis literature with the OSS development context. The implications of the OSS development context on usability cost-benefit considerations have not been addressed. Our analysis revealed that there were interesting parallels and differences when considering the costs and benefits of usability in the OSS development context. The analysis necessitated introducing a refined categorization and interpretation of usability costs and benefits with further implications both for theory and practice. Particularly community OSS development necessitated considerable modifications to the existing models, since this environment has fundamentally different conceptions of what is a 'cost' and a 'benefit'. Regarding limitations, the results are based on the analysis of one case in both community and company OSS development. More cases will be included in the analysis in the future. In addition, implications for OSS use context are not covered in this study, but will be addressed in the future research. It is assumed, that these analyses may lead even to a refined conceptualization of usability cost-benefit analysis.

Acknowledgements

References

Aberdour, M. (2007)Achieving Quality in Open Source Software, IEEE Software 24,1, 58-64.

Andreasen, M., Nielsen, H., Schrøder, S. and Stage, J. (2006) Usability in Open Source Software Development: Opinions and Practice. Information Technology and Control 25(3A), 303-312.

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

Bevan, N. (2000) Cost Benefit Analysis version 1.1. Trial Usability Maturity Process. Serco

Bias, R. and Mayhew, D. (eds.) (1994) Cost-Justifying Usability. Academic Press, Boston

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

Dahlander, L. and Magnusson, M. (2005) Relationships between open source software companies and communities: observations from Nordic firms. Research Policy 34, 481-493.

Donahue, G. (2001) Usability and the Bottom Line. IEEE Software, Vol. 18(1), 31-37

Ehrlich, K. and Rohn, J. (1994) Cost Justification of Usability Engineering: A Vendor's Perspective. In: Bias, R., Mayhew, D. (eds.): Cost-Justifying Usability. Academic Press, 73-110.

Fitzgerald, B. (2006) The Transformation of Open Source Software. MISQ 30(3), 587-598.

Frishberg, N., Dirks, A. M., Benson, C., Nickel, S. and Smith, S. (2002) Getting to know you: open source development meets usability. In Extended Abstracts of the CHI 2002, 932-933.

Grudin, J. (1991) Systematic Sources of Suboptimal Interface Design in Large Product Development Organizations. Human-Computer Interaction, Vol. 6(2), 147-196.

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

Iivari, N. (2008): Usability in open source software development – an interpretive case study. In W. Golden, T. Acton, K. Conboy, H. van der Heijden, V. Tuunainen (Eds.): Proceedings of the 16th European Conference on Information Systems, Galway, Ireland, June 9.-11.2008.

Iivari, N. (2009): "Constructing the Users" in Open Source Software Development – An Interpretive Case Study of User Participation. Information Technology & People 22(2), 132-156

Iivari, N., Hedberg, H. and Kirves, T. (2008) Usability in Company Open Source Software Context. Initial Findings from an Empirical Case Study. In B. Russo, E. Damiani, S. Hissam, B. Lundell, G. Succi (Eds.): Proceedings of the 4th International Conference on Open Source Systems, 359-365.

ISO 13407 (1999) Human-centered design processes for interactive systems. International standard.

Karat, C-M. (1994) A Business Case Approach to Usability Cost Justification. In Bias, R., Mayhew, D. (eds.): Cost-Justifying Usability. Academic Press, 45-70.

Klein, H. K. and Myers, M. D. (1999) A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly, Vol. 23(1), 67-94

Lakhani, K. and von Hippel, E. (2003) How Open Source Software Works: "Free" User-To-User Assistance". Research Policy 32, 923–943.

Mayhew, D. and Mantei, M. (1994) A Basic Framework for Cost-Justifying Usability Engineering. In Bias, R., Mayhew, D. (eds.): Cost-Justifying Usability. Academic Press, 9-43.

Nichols, D. and Twidale, M. (2003) The Usability of Open Source Software. First Monday 8(1)

Nichols, D. and Twidale, M. (2006): Usability Processes in Open Source Projects. Software Process Improvement and Practice 11, 149-162.

Niederman, F., Davis, A. Greiner, M., Wynn, D. and York, P. (2006) A Research Agenda for Studying Open Source I: A Multilevel Framework. Communication of the AIS 18, 19-149.

Nielsen, J. (1993) Usability engineering. Academic Press, Boston

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

Rajala, R., Rossi, M., Tuunainen, V. and Korri, S. (2001) Software Business Models, A Framework for Analysing Software Industry. In Finnish National Technology Agency Technology Review

Rajanen, M. (2007) Usability Cost-Benefit Models – Different Approaches to Usability Cost Analysis. In proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS)

Rajanen, M. (2006) Different Approaches to Usability Cost-Benefit Analysis. In Remenyi, D., Brown, A. (eds.): Proc. ECITE 2006. Academic Conferences, Reading, 391-397

Rajanen, M. (2003) Usability Cost-Benefit Models – Different Approaches to Usability Benefit Analysis. In proceedings of 26th Information Systems Research Seminar In Scandinavia (IRIS26)

Rajanen, M. and Iivari, N. (2007) Usability Cost-Benefit Analysis: How Usability Became a Curse Word? In C. Baranauskas, P. Palanque, J. Abascal, S. Barbosa (Eds.): Proc. INTERACT 2007, Lecture Notes in Computer Science 4663, 511-524.

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

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

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

Ye, Y. and Kishida, K. (2003) Toward an Understanding of the Motivation of Open Source Software Developers. In Proc. 25th International Conference on Software Engineering (ICSE), 419-429.

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

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