Applying usability cost–benefit analysis – explorations in commercial and open source software development contexts

Mikko Rajanen

Mikko Rajanen
mikko.rajanen@oulu.fi
PhD thesis for PhD degree in Information Processing Science in University of Oulu
Department of Information Processing Science
University of Oulu
PO Box 3000
FIN-90014 University of Oulu

Abbreviations

HF	Human Factors
IS	Information System
IT	Information Technology
OSS	Open Source Software
SW	Software
UCD	User-Centered Design
UE	Usability Engineering
UI	User Interface

List of original publications

This thesis is based on the following publications, which are referred to in the text by their Roman numerals.

I Rajanen M & Jokela T (2004) Analysis of Usability Cost-Benefit Models. Proceedings of the 12th European Conference on Information Systems (ECIS2004), Turku, Finland. ISBN: 951-564-192-6.

II Rajanen M (2006) Different Approaches to Usability Cost-Benefit Analysis. Proceedings of 13th European Conference on Information Technology Evaluation (ECITE 2006), Genoa, Italy. ISBN: 978-1-905305-32-2 (book) / 978-1-905305-34-6 (cd)

III Rajanen M & Iivari N (2007) Usability Cost-Benefit Analysis: How Usability Became a Curse Word?. Proceedings of the INTERACT 2007. Rio de Janeiro, Brasil. ISSN: 0302-9743, LNCS: 4663, ISBN: 978-3-540-74799-4, DOI: 10.1007/978-3-540-74800-7_47

IV Rajanen M, Iivari N & Anttila K (2011) Introducing Usability Activities into Open Source Software Development Projects – Searching for a Suitable Approach. Accepted to Journal of Information Technology Theory and Application 2011.

V Rajanen M & Iivari N (2010) Traditional Usability Costs and Benefits - Fitting them into Open Source Software Development. Proceedings of 18th European Conference on Information Systems (ECIS 2010).

Contents

1 Introduction

This chapter establishes the background and motivation of this research, identifies the research questions, gives an overview of the research process and limitations of the research, introduces the cases, defines the scope of the thesis, and gives an overview of the structure of the thesis.

1.1 Background and motivation

> *"When simple things need labels or instructions, the design has failed" (Norman 2002).*

Today, we use software more than ever before, and it plays a crucial role in our everyday and working life. Therefore, it is more important than ever to ask why the software that we use in our daily lives is usually so difficult to use. Why do we have to struggle with incomprehensible user interfaces, and why do we seem to spend more time fighting against the software than using it for something productive? When two users of the same software meet, sooner or later they start sharing their personal horror stories about how the poor user interface made their lives miserable and their work unproductive. Too often, the software or information system is difficult to learn, frustrating to use, its logic of operations follows neither rhyme nor reason, and it does not fulfill the needs of the end user or the organization using it. Poor usability, and hence a stressful work situation, is still a severe problem in computer-supported work, despite efforts to solve these issues (Cajander et al. 2006).

Fortunately, there has been a great deal of progress in regard to making usability improvement activities an integral part of the software development process, and now software development companies usually acknowledge the importance of usability. However, too often the usability improvement activities are the first to be sacrificed whenever there is a need to put the product on the market as soon as possible. Sometimes usability improvement activities are seen as just an additional and maybe even an optional task in software development projects, and in the eyes of the management, such an 'extra' task is always a potential risk for project deadlines. Sometimes poor usability is defended by the argument that the users can be trained and that sooner or later, these users will learn to overcome the usability problems in the system and adapt their work flow to the intricacies of the software. However, it is also possible that the users simply refuse to learn to use the system with poor usability, and that the functionality that has been implemented in the system with so much cost and effort is never used (Zetie 1995). All the innovative and creative technical solutions are in vain if the users cannot easily access them.

Bringing usability activities into the commercial development life cycle has been a challenge since the beginning of the usability activities over fifty years ago (Ohnemus 1996). One reason for these difficulties is that the benefits of better usability are not easily identified or assessed. Usability activities have been competing for resources against other stakeholders in the SW development projects that do have objective and convincing cost-benefit data available for management decision making when the resources are allocated (Karat 1994). Justifying the costs and identifying the benefits of the usability improvement activities have been seen as challenges for bringing usability activities into software (SW) development projects (Bias & Mayhew 1994). Bringing

usability into commercial software development is still a challenge (cf. Ohnemus 1996, Iivari 2006, Rajanen & Iivari 2007).

The open source software (OSS) development context has gained increasing interest in Information Systems (IS) research in recent years (c.f. Fitzgerald 2006, Niederman *et al.* 2006). Initially, SW development was conducted mainly as in-house development where the users and their needs were close and well known. The increasing importance of commercial SW development brought new challenges as users and their needs were no longer as well known to the developers as they had been before. Also, the users were no longer as readily available for usability testing as they had been in in-house development. Traditional in-house development projects have a well-defined user population, participating as a part of their work (Gulliksen 2000). Initially, SW companies paid little attention to OSS development; however, in recent years, it has gained increasing interest from the SW companies. It has been argued that OSS development will be highly influential in the future software landscape (Fitzgerald 2006). Open source software means software with a freely available source code for everyone to access, read, modify, and compile. The fundamental idea of OSS is to enable software to evolve outside of restricted commercial closed source software development in the SW company context by exploiting community participation by technically oriented contributors and users (cf. Raymond 1999, Ye & Kishida 2003). The OSS development context has been compared to a bazaar where people come and go, sell and buy (Raymond 1999). Commercial software development is carried out in closed settings as closed source software development, where only few people can access the code. In commercial software development, the end users can only see and use the binary form of the software. OSS development also makes it possible for the end users to adapt the software to their personal needs and to fix defects (Raymond 1999). OSS development is usually done by technically very skilled developers for their own use, but nowadays OSS solutions have more and more users who have no deep technical knowledge.

Estimating the overall influence of the OSS solutions is difficult because they can usually be downloaded freely and from numerous mirror sites and peer-to-peer networks. Some sources have estimated that the adoption of OSS resulted in savings of about $60 billion to consumers in 2008 and identifies the value of these OSS products to be about 6% of the total value of the software in the world (Standish Group 2008). SourceForge is one of the most well-known web-based repositories and a leading resource for OSS development and distribution. With about 2.7 million developers and over 260.000 OSS development projects, the total number of users in all projects combined is estimated to be more than 46 million, and there are more than two million downloads from project repositories every day (SourceForge.net). There are over twenty other source code repositories and resources for OSS development and distribution. The size of an OSS development project varies from one developer coding and using the application alone to massive OSS development projects spanning years and having hundreds of developers, e.g., Linux and OpenOffice.org. The latest version of OpenOffice.org office application suite has an estimated 15-20% market share. OpenOffice.org announced in 2009 that the latest version had, within a year, recorded one hundred million downloads from their main download site, excluding all downloads from mirror sites and peer-to-peer networks (OpenOffice.org). The Firefox web browser has been downloaded over one billion times and has a 23% worldwide usage share of web browsers (Mozilla.com). The Apache web server software serves 55% of all web sites in the world and has reached the 100 million web site milestone (Apache.org). Therefore, the OSS phenomenon is clearly highly influential in the current software landscape (cf. Fitzgerald 2006).

In OSS development, the solutions tend to be somehow useful in any case since the developers are motivated to develop this particular solution, software, or tool for their own use; but usability has not traditionally been their major concern, partly because, in any event, OSS developers can use the tool that they have been developing mainly for themselves, and partly because OSS developers usually do not have much prior knowledge about usability improvement theory, processes, guidelines, or methods. However, the current OSS usability research is motivated by the fact that there is an ever-increasing number of OSS solutions with a user population that no longer consists only of developer-users. For example, most of the users of some of the most popular OSS solutions, such as Firefox and Apache, are not able to adapt the software to their needs or to fix or report defects (Giuri *et al.* 2004). Improving OSS usability and bringing usability activities into OSS development have not been researched very much, but as more non-developer users have started to use OSS, the importance of bringing usability activities into OSS development and, therefore, improving OSS usability in general is raised. However, the current status of usability activities in OSS projects and the usability of OSS still tends to be quite poor (e.g., Cetin *et al.* 2007, Nichols & Twidale 2003, Nichols & Twidale 2006, Zhao & Deek 2005, Zhao & Deek 2006) even though some recent studies have also identified many usability activities that have already been used in OSS projects (Andreasen *et al.* 2006, Bach & Carroll 2009, Bach *et al.* 2009, Terry *et al.* 2010). Such good progress appears, however, to be rather slow because most OSS core developers are mainly technically oriented and there is a lack of skilled and available usability specialists for OSS development projects. Furthermore, even if there were such usability specialists available, the problem would be to identify and find the OSS development projects that are in need of usability improvement activities and to gain access to the OSS development projects and plan and conduct the usability activities in such a way that they have an impact on the development. There are OSS development projects in need of usability expertise and usability specialists willing to contribute to such projects; but unless the OSS development projects realize they need to integrate these usability improvement activities into their development roadmap, and unless the usability specialists find these projects and find a way to convince the core developers of the importance of usability, these two worlds will never fully meet.

Companies have also currently started to use different forms of OSS in their business and operations. Using and utilizing OSS applications and development tools has been common for a long time, but the utilization of the actual source code is also becoming popular. The availability of free and ready-made components can reduce the development costs substantially, and the OSS development projects and their communities often provide frequent updates and support for these components. Companies have also started to participate in OSS communities and even to launch and build new communities for their products (Dahlander & Magnusson 2005, Fitzgerald 2006, Iivari *et al.* 2008, Niederman *et 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 and implementation (Rajala *et al.* 2001). Some of the recent studies (c.f. Ågerfalk & Fitzgerald 2008) also suggest that outsourcing to the OSS community provides a significant opportunity for SW companies to headhunt top developers from the OSS projects.

The usability cost-benefit analysis models outline the potential costs and benefits of better usability through usability improvement activities (e.g., prototyping, usability testing, and heuristic evaluation) and these models can be used to motivate the management to allocate resources for these usability activities when they see that the

potential benefits outweigh the costs. Though there are some published usability cost-benefit analysis models for the commercial SW development context, the issue of whether these models are really helpful when applied in commercial or OSS development contexts has not yet been studied.

1.2 Research questions

The overall research aim of this thesis is to identify and explore *if usability cost-benefit analysis is helpful when applied in commercial and open source software development contexts*. This broad topic is approached through three research questions:
RQ1: What are the differences and commonalities of the existing usability cost-benefit analysis models?
RQ2: How do the existing usability cost-benefit considerations fit into practice in the commercial development context?
RQ3: How do usability costs and benefits fit into the open source software development context?

These research questions must be examined in numerical order. First, the existing usability cost-benefit analysis models need to be evaluated in order to map out the identified usability costs and benefits from the literature; their underlying assumptions for the usability cost-benefit analysis; and their inherent strengths, weaknesses, differences, and commonalities. Second, these identified usability costs and benefits need to be evaluated regarding how well they fit into practice. Third, the fit of usability costs and benefits into the open source software development context needs to be evaluated and the usability costs and benefits modified to fit the OSS development context.

RQ1: The existing usability cost-benefit analysis literature and research focus on certain areas and have specific viewpoints or different characteristics (c.f. Rajanen 2003, Rajanen & Jokela 2004, Rajanen 2006, Rajanen 2007, Rajanen & Iivari 2007). It is important to take a closer look at the existing usability cost-benefit analysis literature to find out the characteristics and scope of the identified usability costs and benefits. The aim of the first research question is to examine the characteristics of the existing usability cost-benefit analysis literature, the extent to which the existing research identifies usability costs and benefits, and the possibilities of building better universal usability cost-benefit analysis models in general and for the OSS development context in particular. This research question is relevant to the usability cost-benefit analysis research community. Paper I explores the different aspects of usability cost-benefit analysis models and the extent of concrete guidance provided in each model for the cost-benefit analysis. Paper II explores the approach of usability cost-benefit analysis models, the identified empirical background for models, and the identified interest groups.

RQ2: The aim of the second research question is to examine how well the existing usability cost-benefit analysis literature fits into practice in closed source software development and OSS development contexts. It has been argued that usability cost-benefit arguments can be used to promote bringing usability activities into the development process (c.f. Bias & Mayhew 1994). This research question is relevant for researchers doing usability cost-benefit analysis research and practitioners utilizing usability cost-benefit considerations in the company development context. Paper III explores using usability cost-benefit considerations in a case organization in the company development context.

RQ3: It can be argued that the existing usability cost-benefit literature cannot be applied directly in the open source software development context (c.f. Rajanen & Iivari 2010). Therefore, we have to first identify the parallels and differences between open source software development and closed source software development in order to identify the extent to which the existing usability costs and benefits can be directly applied to the OSS development context, and if and how the rest of the usability costs and benefits can be modified to fit them into the OSS development context. It is difficult to introduce usability activities into OSS development because there are no established usability practices or culture, and core developers are not familiar with usability as a concept and often do not see any need for it (c.f. Rajanen, Iivari & Anttila 2011). It has been argued that usability cost-benefit arguments can be used by management to promote bringing usability activities into the development process (c.f. Bias & Mayhew 1994, Bias & Mayhew 2005). The OSS development context-specific usability costs and benefits are reflected in the experiences of bringing usability into the OSS development context (c.f. Rajanen, Iivari & Anttila 2011). This research question is relevant for usability researchers, OSS researchers, usability cost-benefit analysis researchers, and usability advocates in the OSS development context. Paper IV explores bringing usability into the OSS development context. Paper V explores the implications of OSS for usability cost-benefit considerations and introduces a specific prototype usability cost-benefit analysis model that could be helpful in the OSS development context.

1.3 Cases

Different development contexts have been identified. Grudin (1991) identifies three different development contexts: the in-house, tailored, and product development contexts. In the in-house development context, the SW is developed for use within the development company itself. In the tailored or contract development context, the SW is developed for a particular customer organization and the SW is tailored to fit the needs and demands of the customer. In the product or commercial development context, the SW is developed for any potential customer with no particular tailoring.

1.3.1 Commercial development context

This thesis presents one case of using usability cost-benefit arguments in the company development context. The company case organization was a small to medium-sized SW development company with not much prior knowledge about usability or user-centered design (UCD), with a typical organizational hierarchical structure, and developing large-scale business-to-business information systems and SW-intensive products targeting international markets. The company operated in the tailored development context. The SW was developed for a particular customer and tailored to fit the specific needs and demands of that specific customer in a unique context of use. Initially, the management at the company and project level was very committed to the usability improvement efforts. The company was chosen as a case organization because it had a limited background in usability work and there was open access to top and project management to introduce usability cost-benefit considerations. Access to this case project was gained through a research project that aimed to introduce usability activities into commercial SW development organizations. The case organization participated in the research project for two years. The usability improvement effort was initiated by evaluating the current state of the usability activities in the organization. These usability improvement efforts

included experimenting with many different kinds of usability activities, such as paper prototyping, usability testing, UI style guide development, and customer visits. Usability cost-benefit considerations were introduced to top- and project-level management as arguments for these usability improvement efforts.

1.3.2 OSS development context

This thesis presents six cases of bringing usability activities into open source software (OSS) development projects. The OSS case projects involved in this thesis were OSS development projects with different sizes, levels of organization, and development aims. These cases were studied to identify possible areas of usability cost-benefit considerations in the open source software development context, to identify the similarities and differences between closed source software development and these open source software development projects, and to pilot test the identified usability cost-benefit arguments for open source software development. These OSS development projects were developing software for non-technical end-users. Some projects developed software for very large and wide end-user target populations, while other projects had very small but highly specialized and skilled target end users.

The OSS development projects had a long development history, and projects had enough core developers and active community members to ensure that the projects remained active during the studies. Also, this made the research setting more authentic since new contributors join an OSS development project when the project has already progressed beyond the initial design phase and the first versions of the software have already been released. Most of the projects were small enough to identify the core developers and to communicate directly with them; but not too small. The projects had great differences in their hierarchical structure, decision-making process, means of communication, and development rigor. The case projects were deliberately chosen to not include any projects with some type of official SW company participation or endorsement. This exclusion was made because companies may have usability resources and may prefer to utilize them in the development if they participated in or endorse the development (e.g., Benson *et al.* 2004, Frishberg *et al.* 2002, Iivari *et al.* 2008, Nichols & Twidale 2006). Also, any OSS project that has been started by an SW company or is closely monitored and guided by an SW company may not be very different from a closed source software development project within the company from the usability activities point of view; therefore, while it might offer an interesting area for further research, it was excluded.

Access to these six case OSS development projects was gained through six student teams that aimed to introduce usability activities into their selected OSS development project under close supervision of the author. The students in these usability teams had a usability background from at least two previous usability courses about usability evaluation methods (e.g., heuristic evaluation and usability testing), user-centered design and user interface design in both theory and practice. Each student team consisted of three to five students working 200 to 230 hours each in planning the usability activities, carrying out the usability evaluation and improvement activities, collecting data, communicating the evaluation results and suggested UI improvements to the OSS development project, and writing a project report.

The student teams introduced usability activities into their selected OSS development project in a way that was based on the experience and guidance of the author and collected data related to these usability activities and to the history, structure, and culture

of that particular OSS development project. The author supervised and guided the usability activities introduced by the student teams, analyzed the impact of usability activities on the case projects, and made research assumptions for the following cases.

1.4 Structure of the thesis

This thesis has been divided into the following parts: Introduction, Related work, Research process and procedure, Analysis and synthesis, and Summary.

Related work (Chapter 2) establishes the theoretical background for the thesis, links it to the existing research in this area, and presents the key concepts and definitions in this research area.

Research process and procedure (Chapter 3) establishes the research process and outlines the procedures of the research.

Analysis and synthesis (Chapters 4 and 5) establishes the answers to the research questions. First, the differences and commonalities of the existing usability cost-benefit analysis models are identified and analyzed (Chapter 4). Second, these usability cost-benefit analysis considerations are contrasted with the commercial SW development in order to estimate how well they fit into the commercial SW development context. Third, the existing usability cost-benefit considerations are modified so that they can be applied in the open source software development context (section 5.3). Fourth, these modified usability costs and benefits are contrasted with the OSS development context in order to estimate how well they fit into open source software development projects.

Summary (Chapter 6) presents the results of the research, a final summary of the research, and the contribution of this research to the research and practice; identifies the limitations of this research; identifies areas for future research; and discloses the research and results of the thesis.

2 Related work

This chapter establishes the theoretical background for the thesis, links it to the existing research in this area, and presents the key concepts and definitions in this research area.

2.1 Usability and usability cost-benefit analysis

> *"E-commerce shifts the emphasis from the advantages of being usable to the penalties of not being usable" (Hughes 2002).*

The development context affects usability and usability cost-benefit considerations. There are also studies that indicate that the type of development context has an effect regarding usability, user-centered design activities and level, and ease of user involvement (Iivari & Molin-Juustila 2009). For example, the benefits of better usability of SW developed for in-house use can be identified within the development organization (c.f. Bevan 2000), reduced development and support costs can be seen as having more impact when SW is tailored for specific customers, and increased sales can be seen as having more impact in the product development context.

2.1.1 Usability

> *"Would somebody please think of the users?" (Author)*

Usability is defined as one of the main SW product and system quality attributes in the international standard ISO 9126. In this standard, usability refers to the capability of the product to be understood, learned, used by, and attractive to the user, when used under specified conditions (ISO 9126). The second common definition for usability is in standard ISO 9241-11, where usability is defined as being 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 9241). The third common usability definition is by Nielsen and Schneiderman, who define usability as consisting of five quality components: learnability, efficiency, memorability, errors, and satisfaction (Nielsen 1993, Schneiderman 1998). In this thesis, usability is defined as in the ISO 9241-11 standard and the terms usability engineering (UE), human factors (HF), usability work, usability activities, and user-centered design are used synonymously. Usability can be achieved through a user-centered design process, usability activities (e.g., usability testing, paper prototyping, heuristic evaluation), and having an overall focus on usability issues through the entire development process (c.f. Boivie *et al.* 2005, Gulliksen *et al.* 2003, Gould 1997, Landauer 1995).

2.1.2 Cost-benefit analysis

Cost-benefit analysis is a method for assessing the projects from the investment point of view (Karat 1994). This method is based on making investment decisions by comparing the estimated costs and benefits of the planned actions. This comparison is based on collected and analyzed data regarding technology and finance. This comparison helps the management to focus the available resources in the most useful way on such planned activities that have low costs and potentially high benefits. While there are plenty of different cost-benefit models for different contexts ranging from restricting carnivorous plants (Givnish *et al.* 1984), rural to urban migration (Speare 1971), and to electronic medical records (Wang *et al.* 2003), there are relatively few published models for

analyzing the costs and benefits of usability in the company development context. In a literature search of HCI journals, ACM and IEEE digital libraries, and top conference proceedings from 2005 to 2011, no similar interpretive case studies were found from the mainstream HCI research where usability cost-benefit analysis models would have been used in an empirical setting, or where the results of using usability cost-benefit considerations in an empirical setting would have been contrasted with the existing usability cost-benefit analysis literature.

The usability cost-benefit analysis models explored in this thesis from the book *Cost-Justifying Usability* (Bias & Mayhew 1994) are by Ehrlich and Rohn, Karat, and Mayhew and Mantei. The second edition of the book was published in 2005 and it did not change the usability cost-benefit models but rather had a specific focus of applying usability cost-benefit considerations to web and intranet contexts (Bias & Mayhew 2005). In addition, the thesis explores usability cost-benefit analysis models by Bevan (2000) and Donahue (2001). These usability cost-benefit analysis models are described in more detail in Paper I and are very different in their categorization of the usability cost and benefits and the amount of provided guidelines and identified empirical background for the usability cost-benefit analysis.

The usability cost-benefit analysis method has three steps and it proceeds as follows (Burrill & Ellsworth 1980):

1. Identify the financial value of expected project cost and benefit variables.
2. Analyze the relationship between expected costs and benefits using simple or sophisticated selection techniques.
3. Make the investment decision.

Cost in the cost-benefit analysis context means the estimated or projected monetary or abstract expense of doing a particular action or starting a particular project. This cost can be concrete and, therefore, easily measurable and quantifiable, or abstract and, therefore, difficult to measure or quantify in financial terms (Burrill & Ellsworth 1980). The objective is to find more or less accurate financial estimates for each of the costs, be they concrete or abstract by nature. Sometimes it may be impossible to estimate a reliable financial impact for the abstract costs. In these cases, the best estimate or the range of various estimates is presented and the inaccuracy of the estimation is taken into account when the costs are analyzed.

In an SW development project, the typical concrete costs are direct project expenses (e.g., salaries of the project personnel and expenses of project offices), one-time purchases (e.g., equipment and software), one-time deployment costs (e.g., reduced productivity due to implementing new technology) and continuous overall expenses (e.g., maintenance and support for offices and equipment, training of the personnel). In addition to these concrete costs, there can be many kinds of abstract costs impacting the project. For example, the problems in knowledge transfer due to high staff turnover will cause the project indirect costs that are very difficult to estimate and quantify in monetary terms. However, even these kinds of difficult-to-estimate and abstract costs should be taken into account, even if the exact monetary value cannot be evaluated by any reasonable means (DIRKS 2003).

Benefit in cost-benefit analysis is an expected positive result of the planned action or project through either cost saving or estimated added value (Burrill & Ellsworth 1980, DIRKS 2003). Just like costs, benefits can be divided into concrete and abstract categories. In any kind of project, typical concrete benefits can be divided as improved

productivity (e.g., due to smaller expenses or when available resources are used more efficiently), improved effectiveness (e.g., by optimizing the provided services) and indirect benefits (e.g., using the analysis for process improvement) (DIRKS 2003). Abstract benefits in any kind of project or organization might be, for example, improved customer loyalty or the increased reputation of the company. These kinds of abstract benefits are extremely difficult to evaluate or quantify in monetary terms; in fact, it is argued that the amount and impact of the improved reputation of the organization cannot be measured in any sensible way (Due 1989).

2.1.3 User-centered design

"Making the world a better place one user interface at a time." (Author)

In the early days of information technology, the SW developers were usually themselves the users of the SW they wrote and, therefore, knew their own needs and the context of use. Now IT solutions are used everywhere and users can be of any age, from any culture, or from any context imaginable. The SW developers no longer have direct knowledge about the tasks, skills, and experience of the users, or information about the context within which the system is used. This can result in SW that does not answer to the functional needs of the user, but has plenty of extra unnecessary features, and does not sit well in the established work process of the user. SW developers need information about users and the context of use. UCD has been developed to fulfill this role. According to international standard ISO 13407, UCD can be described as consisting of four principles and four iterative design activities. The four principles of user-centered design identified in ISO 13407 are:

- Active involvement of users in design activities
- Iterative design where design solutions are produced in iterative and incremental fashion
- Multi-disciplinary design where skills and views of people with various backgrounds are utilized in the design
- Allocation of tasks and functions to system and to user where appropriate

The four user-centered design activities need to start at the beginning of a development project in iterative fashion, and these activities identified in ISO 13407 are:

- Understanding and specifying the context of use
- Specifying the user, task, and organizational requirements
- Producing design solutions
- Evaluating designs against requirements

The iterative and incremental user-centered design process continues until the design solution is evaluated as fulfilling all the user and organizational requirements (ISO 13407).

The user-centered design process may include several usability improvement actions. In understanding and specifying the context of use and specifying the user and organizational requirements, these usability improvement actions can be, for example, creating personas, customer visits, and usability requirement workshops (c.f. Gulliksen *et al.* 2003). In producing design solutions and evaluating those against requirements, the usability improvement actions can be in the form of paper prototyping, expert evaluation, usability testing, and so on (Gulliksen *et al.* 2003).

One recent study has examined user-centeredness in the systems development context from the viewpoint of the four principles of user-centered design. This study found that there are considerable variations in how four allegedly user-centered systems design methods address the four principles of user-centeredness (Iivari & Iivari 2010).

2.2 OSS development

OSS is computer software that is freely available as source code—and often also as a precompiled binary file—where the license permits the users to read, change, and modify the source code as derived works, recompile the modified source code to binary form, and distribute the modified source code as a derivative under the same license as the original source code. The fundamental idea of OSS is to enable software to evolve freely by exploiting community participation. OSS also makes it possible for end users to adapt software to their personal needs and fix defects (Raymond 1999).

There are different kinds of licenses used in OSS such as GNU General Public License (GPL), GNU Library General Public License (LGPL), and BSD Licenses. OSS development has gained interest in Information Systems (IS) research in recent years (see, e.g., Fitzgerald 2006, Niederman *et al.* 2006) and OSS development has been argued to be highly influential in the future software landscape (Fitzgerald 2006).

Usually, the OSS is developed as a public and collaborative effort in the OSS development project with core developers, developers, contributors, and users. The OSS communities are often depicted with an onion model with different layers representing the level of involvement within that particular OSS community (cf. Aberdour 2007).

2.2.1 Historical and cultural background of OSS

Software communities that can be compared to modern free/open software communities have existed for a long time before the free software movement and before the term "free software" was coined (Levy 1984). The free software movement was launched in 1983 as a social and political movement to advocate what was seen as the basic freedoms of software users. These freedoms were identified as the freedom to run the software, freedom to study the software, freedom to change the software in any way that the user sees as necessary, and the freedom to distribute copies of the software with or without changes to it. These freedoms are seen as promoting the progression of technology since much of the wasteful duplication of programming efforts could be avoided and these efforts can instead go into advancing the state of the art (Stallman 1985, referenced in Wardrip-Fruin & Montfort 2003). A reflective analysis of the hacker culture and free software principles and communities by Raymond (1997) motivated Netscape to release its internet browser as free software. This was the starting point of the popular Mozilla Firefox internet browser and the Thunderbird email client. The term "open source" was coined to rebrand the free software movement so that it would be more appealing to the commercial software industry in order that it would see the benefits of the open development process. The Open Source Initiative was founded in 1998 to promote this new term and to advocate the open source principles (opensource.org). The members of the free software movement objected to this open source approach and felt that by concentrating only on the openness of the source code, the important philosophical and social values about the basic freedoms of the software users were ignored (gnu.org). Despite the differences in some ideologies, the OSS and free software communities share many of the core values, such as placing a high value on freedom of speech, regarding

programs as communal resources, and considering free information sharing as a right and an ideal (Rolandsson *et al.* 2009, Himanen 2001, Szczepanska *et al.* 2003). Also helping others so that they may solve new problems instead of readdressing old ones, and technical knowledge, skill, and learning for its own sake are common values for both communities (Rolandsson *et al.* 2009, Raymond 2003, Stewart & Gosain 2006).

2.2.2 OSS development community

An OSS development project is a loosely coupled community kept together by strong common values such as that software should be free and work is kept together by one or a few coordinators (Ljungberg 2000). The OSS community is often depicted with an onion model with different layers representing the level of involvement within that particular OSS community. In a typical OSS community, there is a lead developer or a small group of developers forming a core team that controls the overall architectural design and the course of the project (Feller & Fizgerald 2000, Mockus *et al.* 2000). These project leaders making decisions in the project form the hard core of the onion. These project leaders are often 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 errors in the code. Although these contributors can download and modify the source code, they do not have the power to upload their modifications to the official source code repository of the project. The outer layer of the onion consists of end users who do not participate in the community, but only use the software (Aberdour 2007). The level and structure of organization varies between different OSS development projects but usually small and medium-sized OSS development projects have an informal, shallow, and meritocratic organizational structure where the contributors, whose contribution is seen as being important or innovative, are often given developer or core-developer status by the decision of the developers or community as a whole. One of the main motivational forces in the community is the status, fame, reputation, and recognition that a contribution can give to the developer (Raymond 1999, Aberdour 2007, Bergquist & Ljungberg 2001, Berquist 2003, Zeitlyn 2003).

2.2.3 OSS developers as users

Technically skilled developers have traditionally developed OSS for their own use, but OSS solutions now have increasingly more users who lack in-depth technical knowledge. The OSS solutions have tended to be useful since the developers have produced tools that they themselves use and, therefore, know the user, task, and organizational requirements of those tools, as well as their context of use; however, usability has not traditionally been their major concern. There is now an increasing number of OSS solutions that have user populations that no longer consist solely of developer-users; most of the users of several popular OSS solutions, such as the popular web browser Firefox, may not be capable of adapting the software to their needs or fixing or reporting defects like developer-users could (Giuri *et al.* 2004). However, some studies indicate that the OSS community is starting to acknowledge that ‘we are not our users,’ usability of the OSS is poor, and the OSS development process should be characterized as anything but user centered (Frishberg *et al.* 2002, Benson et al. 2004, Nichols & Twidale 2006, Pemberton 2004, Zhao & Deek 2005, Zhao & Deek 2006).

Improving OSS usability and bringing usability activities into OSS development have not been researched in detail until recently, but as more non-developer users start to use OSS, the importance of these factors will increase. A similar phenomenon can be identified when non-technical users first started using computers and software they had not developed themselves, and the difficulties using the systems and software they had not developed was the impetus for the first usability studies. A similar trend can also be seen in web interfaces where, first, only the technically savvy people could use or develop web pages. Even though recent studies have described some usability activities that have already been carried out in some OSS projects (Andreasen *et al.* 2006, Bach & Carroll 2009, Back *et al.* 2009, Terry *et al.* 2010), the current status of usability activities in OSS projects and the usability of OSS still tend to be quite poor (e.g., Cetin *et al.* 2007, Nichols & Twidale 2003, Nichols & Twidale 2006, Zhao & Deek 2005, Zhao & Deek 2006), and this is still the case, especially in small and medium-sized OSS projects where the developers typically do not have theoretical or practical knowledge about user interface and usability methods and practices, and in OSS projects without company involvement. When a company is involved in an OSS project, it usually brings its UI and usability methods and practices into the development roadmap.

2.2.4 OSS developers as managers

One of the most common models in coordinating OSS development projects is that several contributors work under a single 'benevolent dictator' who is usually the founder of the project and who attracts committed and talented contributors (Ljungberg 2000). An alternative to having one single benevolent dictator is rotating dictatorship or forming a voting committee from the developers (Ljungberg 2000). To a certain extent, the core developers can be argued to have the same characteristics as managers in commercial software development. Core developers make both low-level decisions regarding whether or not a particular contribution is accepted to the code, mid-level decisions regarding software features to be included in individual releases, and strategic decisions regarding the direction of the development in the future as well as the development roadmap.

However, some clear differences are also apparent when comparing OSS core developers to the managers in commercial SW development. First, the core developers in the OSS development context do not usually issue tasks for individual developers or issue strict development deadlines. Second, in most OSS projects, the core developers do not have to budget and allocate limited human, technological, or financial resources. The managers in commercial software development have to balance the development activities within the overall resources allocated to them. Since all usability activities require some kind of resources, management support is important when trying to bring usability activities to commercial software development. This is why the traditional usability cost-benefit analysis models focus heavily on gaining management commitment and support by identifying areas such as the lesser need for resources as a result of usability activities. This approach is not directly suitable for advocating usability activities in OSS projects because of the lack of budgeted resources and, therefore, the OSS context needs its own tailored usability cost-benefit aspects in order to catch the interest of OSS core developers who are not concerned about time or resources (Rajanen & Iivari 2010).

3 Research approach and methods

3.1 Research approaches

The research process of this thesis utilizes conceptual-analytical and constructive research approaches (March & Smith 1995, Järvinen 2000). In IS research, the design science approach is a type of scientific research that aims to develop new or improved ways to achieve human goals (c.f. March & Smith 1995, Hevner *et al.* 2004). Design science consists of two basic activities: building and evaluating. Building is a process for constructing an artifact for a specified purpose, and evaluating is a process of determining how well the constructed artifact performs in that specified purpose (March & Smith 1995). Building and evaluating IT artifacts has design science intent (March & Smith 1995). In this thesis, Paper V has the perspective of building a usability cost-benefit model to fit into the OSS development context and Paper IV has the perspective of evaluating these OSS usability cost-benefit considerations. March and Smith differentiate two cases concerning when the construct already exists in some form and when the construct has not existed before in any shape or form. In case the construct is totally new, the contribution of the research comes from the novelty of the artifact and the persuasiveness of the claims that it is effective (c.f. March & Smith 1995, Järvinen 2000). In case the construct has already existed in some form, the contribution of the research lies in the new form of the construct being, in some sense, better than the old one (c.f. March & Smith 1995, Järvinen 2000).

Case study research is an empirical enquiry in which the focus is on a contemporary phenomenon in its real-life context when the boundaries between the phenomenon and its context are not clearly evident, and in which multiple sources of evidence are used (Yin 1994). The case studies may be explanatory, exploratory, or descriptive and can be either qualitative, quantitative, or both (Yin 1994). An explanatory case study is used to study and explain a particular case and find its underlying principles. An exploratory case study is used to study the case in depth. A descriptive or interpretive case study tries to understand and make sense of the case and not to explain it in a predictive sense. However, there is no exclusivity between explanatory, exploratory, and descriptive case studies (Yin 1994). In case study research, the theoretical propositions are used as sensitizing devices to guide the collection and analysis of the data and typically multiple sources of data are preferable (Yin 1994).

Table 1. Research questions and research approaches.

Research question	Research approach
RQ1: What are the strengths, weaknesses, differences, and commonalities of the existing usability cost-benefit analysis literature?	Conceptual analytical *Understand*
RQ2: How do the existing usability cost-benefit considerations fit into practice?	Case study, qualitative research *Understand*
RQ3: How do usability costs and benefits fit into the open source software development context?	Conceptual analytical, constructive research, case study qualitative research, *Understand, Evaluate*

In this thesis, Papers I, II, and III are focused on understanding the usability cost-benefit analysis in the commercial development context. Paper III also evaluates the usability cost-benefit analysis in the commercial development context. Paper IV and Paper V are focused on understanding the OSS development context and building and evaluating the usability cost-benefit considerations fitting the OSS development context.

Table 2. Papers and research process

	Commercial	OSS
Understand	I, II, III	IV, V
Build		V
Evaluate	III	IV

3.2 Data gathering and analysis

The empirical data in this thesis were gathered from one case in the commercial SW development context and four cases in the OSS development context. The commercial SW development case organization was a small to medium-sized SW development company that was developing large-scale business-to-business IS- and SW-intensive products for international markets. Access to this case organization was gained through a research project attempting to introduce usability activities into SW development organizations. The case project participation and data gathering lasted two years.

Access to the four cases in the OSS development context was gained through four student usability projects: UKKOSS 1, UKKOSS 2, UKKOSS 3, and UKKOSS 4. These student usability projects aimed to introduce usability activities into OSS development projects. The student usability projects communicated with their allocated OSS development project and tried to introduce usability activities into it in order to have a 'wake-up call' (cf. Schaffer 2004) among the core developers and the community regarding the importance of usability.

Table 3. Description of the OSS case projects

	Case 1	Case 2	Case 3	Case 4
Number of developers	~30 developers	~15 developers	~40 developers	~20 developers
Number of users	Medium to large user base (<16.000)	Small user base (<1000)	Very large user base (>800.000)	Very large user base (>800.000)
Application type	Media center software	Game	3D content creation software	Media center software
Starting year	2004	2003	2002	2003
Development activity	Active development	Sporadic activity	Active development	Active development

Decision making	Authoritative (core developers)	Community based	Authoritative (core developers)	Authoritative (core developers)
Usability team	5 students	3 students	10 students	3 students
Usability activities	Expert evaluation, usability testing	Expert evaluation, usability testing, user interface design	Expert evaluation, usability testing, mock-ups	Expert evaluation, usability testing

The students conducting these usability projects had a usability background from at least two previous usability courses about usability evaluation methods (e.g., heuristic evaluation and usability testing), user-centered design, and user interface design in both theory and practice. Each student usability team consisted of three to five students working about 200 hours each in planning the usability activities, carrying out the usability activities, communicating with the OSS project, following up the impact of usability activities, collecting data, and writing a project report. The student projects introduced usability activities into one selected OSS development project, based on assumptions and guidance from the author, and collected data related to these usability activities and some issues related to the history, structure, and culture of the case OSS project. The author supervised and guided the usability activities introduced by the student projects, analyzed the impact of usability activities on the case projects, and made plans for the following cases. Prior to these usability activities, the OSS case projects 1, 2, and 4 had very limited knowledge and almost no background at all in usability activities. The OSS case project 3 had some background in user interface design, but very limited background in implementing usability activities.

The research material has been gathered over the period of five semesters. The material was collected while conducting usability activities and observing OSS development projects. Usability findings and recommendations and all forum and email correspondence between the development teams and the student usability teams have been saved for the purposes of the research. The OSS developer culture places more value on the functionality of code than on interaction design, OSS projects tend to be developed piecemeal by separate individual contributors, and comprehensive design and evaluation efforts do not fit the development process well (Green *et al.* 2009); therefore, it can be argued that the results would have remained the same even if a group of professional usability specialists had tried the same consultancy-style approach.

4 Results: Analyzing the usability benefits

The results of this thesis are presented by summarizing the results of the individual papers in chronological order.

4.1 Analyzing the usability cost-benefit analysis models (ECIS 2004)

Paper I contributes to the thesis by analyzing the different cost-benefit analysis models that are used for estimating the costs and benefits of better usability through usability activities. There exist a number of different cost-benefit models related to usability. However, there is not much work reported on analyzing, contrasting, and comparing these different usability cost-benefit analysis models. The aim of this paper is to partially fill this gap.

The analytical framework for analyzing, contrasting, and comparing the usability cost-benefit analysis models is based on the product life cycle. This makes it possible to conduct a systematic analysis that takes account of the different roles that usability has in different phases. The main phases of the product life cycle are identified as follows:

- Product development phase
- Product sales phase
- Introduction phase
- Daily use

In the first phase, product development, the benefits are actually not based on usability as a measurable product attribute, but rather on user-centered design as a product development paradigm. Usability as a product attribute becomes important at the phase in which the product gets in touch with the end users. In the following, each of the selected models is analyzed from the viewpoints of these four different phases of the product life cycle. The main research questions of this paper are:

- What kinds of aspects of benefits does each model cover at each phase?
- To what extent does each model provide concrete guidance for estimating the benefits?
- How are the steps of the cost-benefit analysis method identified and documented in usability cost-benefit models?

As a last topic of research, the related costs or usability activities are examined. These costs are mainly related to the first product life cycle and its development phase.

The analysis of this paper covers four models, three of which were formulated by Ehrlich and Rohn, Karat, and Mayhew and Mantei, and are presented in the book by Bias and Mayhew (1994). Although the book is rather old, it is still the benchmark in cost-justifying usability and usability cost-benefit considerations. This book has recently been cited in works ranging from the benefits and challenges of user involvement (Kujala 2003), lack of usability cost-benefit considerations in the consumer appliance market (Jokela 2004), explorations in workplace user frustration with computers (Lazar *et al.* 2006), and the engineering design of systems (Buede 2008), to making the design process more usable (Champney *et al.* 2011), and there seem to be no apparent trends in these citations. The second edition of the book was published in 2005 and it did not change the usability cost-benefit models; rather, it had a specific focus of applying usability cost-benefit considerations to web and intranet contexts (Bias & Mayhew 2005). The fourth model included in this paper is by Bevan (2000), and is more recent than other usability cost-benefit analysis models and has a different background since it is a result of

European research projects. There is also literature that does not directly reference these previous bodies of research but use the cost-benefit considerations to, for example, analyze the costs and benefits of discount usability methods (cf. Cockton & Woolrych 2002).

4.1.1 Ehrlich and Rohn

Ehrlich and Rohn analyze the potential benefits of better usability from the point of view of the vendor company, corporate customer, and end user. They state that by incorporating the usability activities into the product development project, both the company itself and its customers gain benefits from certain areas. When compared to other usability benefit models analyzed in this paper, Ehrlich and Rohn present the most comprehensive discussion about different aspects of usability cost-benefit analysis. They do not clearly present an overall formula for calculating the value of usability costs and benefits.

According to Ehrlich and Rohn, the vendor company can identify benefits from three areas:

1. Increased sales
2. Reduced support costs
3. Reduced development costs.

In some cases, the link between better usability and increased sales can be found, but it can often be difficult to relate the impact of better usability directly to increased sales. One way to identify the impact of usability on sales is to analyze how important a role the usability has in the buying decision of the users.

The cost of product support can be surprisingly high if there is a usability problem in an important product feature and the product has lots of users. Better usability has a direct impact on the need for product support and, therefore, great savings can be made through less need for support. By focusing on better product usability and using usability methods, the vendor company can cut development time and costs. The corporate customer can expect benefits when more usable products reduce the time that the end users need to devote to training. In addition to official training, there are also hidden costs for peer support. End users often seek help from their expert colleagues, whose productivity decreases as a result. It is estimated that this kind of hidden support cost for every PC is between $6.000 and $15.000 every year (Bulkeley 1992).

End users are the final recipients of more usable products. According to Ehrlich and Rohn, increased usability can result in higher productivity, reduced learning time, and greater work satisfaction for the end user. The end user can benefit from higher productivity when the most frequent tasks take less time.

4.1.2 Bevan

Bevan estimates the potential benefits of better usability to the organization during development, sales, use, and support. The vendor can gain benefits in development, sales, and support. The customer can benefit in terms of use and support. When the system is developed for in-house use, the organization can identify benefits in development, use, and support. In each category, there are a number of possible individual benefits where savings or increased revenue can be identified.

The total extent of the benefit from better usability can be calculated by adding all identified individual benefits together. Bevan mainly discusses the usability benefits through increased sales, less need for training, and increased productivity. Benefits through decreased development time are identified but are not discussed in detail.

4.1.3 Karat

Karat approaches the usability benefits through a cost-benefit calculation of human factors (HF) work. This viewpoint is different from other analyzed usability cost-benefit analysis models. The following are examples of identified potential benefits:

1. Increased sales
2. Increased user productivity
3. Decreased personnel cost through smaller staff turnover.

The development organization can gain benefits when better usability creates a competitive edge and, therefore, increases product sales. Customer organization can gain benefits when end-user productivity is increased through reduced task time and when better usability reduces staff turnover. Karat describes a usability cost-benefit analysis of three steps. In the first step, all expected costs and benefits are identified and quantified. In the second step, the costs and benefits are categorized as tangible and intangible. The intangible costs and benefits are not easily measured, so they are moved into a separate list. The third step is to determine the financial value of all tangible costs and benefits. Karat also links the usability cost-benefit analysis to business cases. Business cases provide an objective and explicit basis for making organizational investment decisions (Karat 1994).

4.1.4 Mayhew and Mantei

Mayhew and Mantei argue that a cost-benefit analysis of usability is best carried out by focusing the attention on the benefits that are of most interest to the audience for the analysis. The relevant benefit categories for the target audience are then selected and benefits are estimated. Examples of relevant benefit categories are given for the vendor company and internal development organization. The vendor company can benefit from:

1. Increased sales
2. Decreased customer support
3. Making fewer changes in late design life cycle
4. Reduced cost of providing training.

4.1.5 Benefits in the development phase

Overall, the explored usability cost-benefit analysis models identify three different kinds of usability benefits that user-centered design can provide in the development phase: reduced development costs, prioritization of product features, and less need for future redesign.

By focusing on better product usability and using usability techniques, a vendor company can reduce development costs. Ehrlich and Rohn, Bevan, and Mayhew and Mantei identify reduced development costs as one potential benefit. Mayhew and Mantei provide a sample calculation for analyzing this benefit. They calculate the benefits

through comparing the difference between the costs of changes detected early and those detected late. Ehrlich and Rohn have a descriptive discussion but no concrete guidelines or example calculations. Bevan mentions these benefits only briefly, and Karat does not discuss them at all (Table 4).

Ehrlich and Rohn and Bevan discuss, to some extent, the benefit of prioritizatizing the functionality that is important for customers. In one case, three key features were deliberately added to the product to make it more appealing, but 95% of the respondents to a survey never used the features because they did not know that the features existed, did not know how to use the features, or did not understand the features (Ehrlich & Rohn 1994). Karat and Mayhew and Mantei do not address this point specifically.

In addition to these benefits, Bevan identifies the reduced need for architectural redesign to make future versions of a product easier to use as a potential benefit.

Table 4. The extent to which usability cost-benefit models identify and document benefits

Benefit category	Ehrlich & Rohn	Karat	Bevan	Mayhew & Mantei
Reduced development costs	XX	-	X	XX
Prioritization of product features	XX	-	X	-
Less need for future redesign	-	-	X	-
XXX = The benefit is identified and well documented, concrete guidelines, examples, etc. XX = There is some discussion about the benefit, no concrete guidelines X = The benefit is identified - = The benefit is not identified				

None of the analyzed models takes into account the potentially different benefits depending on whether the product is tailored or mass produced. It would be interesting to see whether there are some differences in estimating the benefits of better usability when the product is tailored or mass produced. It is known that different requirements and principles are applied for developing tailored products and mass products. Because none of the models in this study offer different analysis for these two approaches, it would be interesting to see the extent to which the usability benefits are different in these cases. One research study identified user-centered design as having a key role in differentiating product and HF improvements (Harrison *et al.* 1994).

4.1.6 Benefits in the sales phase

Generally, the models identify two categories of usability benefits in the sales phase: gaining a competitive edge, and increased customer satisfaction. It is very difficult to estimate the impact of better usability on product sales. However, there are some reported cases, where a link between better usability and increased sales can be established. In one reported case, revenues grew by 80% when the most serious usability problems were fixed in the second release of a product (Wixon & Jones 1991). Poor usability may have a serious effect on a company's reputation and market share, especially when the market is tightly controlled (Mauro 1994). Also, product development usability can speed up a product's market introduction and acceptance (Conklin 1991).

The benefits of gaining a competitive edge by claiming a product is easier to use than other products is identified and discussed in all models (Table 5). Ehrlich and Rohn have a detailed discussion about this benefit, and Mayhew and Mantei have an example

calculation where the number of systems sold due to enhanced usability is multiplied by the profit margin per product.

Increased customer satisfaction can result in more repeat customers and, therefore, increased sales. Ehrlich and Rohn and Bevan identify the benefits of customer satisfaction and have descriptive discussions about the benefit. Ehrlich and Rohn estimate that satisfied customers influence four other people to buy the same brand, and dissatisfied customers influence ten other people to avoid the brand. Karat and Mayhew and Mantei do not identify this benefit explicitly.

Table 5. The extent to which usability cost-benefit models identify and document benefits for marketing and sales.

Benefit category	Ehrlich & Rohn	Karat	Bevan	Mayhew & Mantei
Gaining a competitive edge	XX	XX	XX	XX
Increased customer satisfaction	XX	-	XX	-
XXX = The benefit is identified and well documented, concrete guidelines, examples, etc. XX = There is some discussion about the benefit, no concrete guidelines X = The benefit is identified - = The benefit is not identified				

One observation from this exploration and analysis is that none of the analyzed models takes into account the benefits of better usability in terms of sales depending on whether the product is a business-to-business or a business-to-consumer product. None of the models discusses this, although it could be argued that there may be differences in estimating the benefits of better usability in the case of business-to-business products vs. business-to-consumer products.

4.1.7 Benefits in the introduction phase

There are two categories of usability benefits for product support: a reduced cost of product support and less need for end-user training. The difference in training time between a usability-engineered system and a system designed without usability engineering (UE) can be as much as several days (Karat 1993). Training the end user includes official training conducted by the development organization or a customer organization and unofficial training by skilled peers.

Ehrlich and Rohn, Bevan, and Mayhew and Mantei identify and discuss the reduced cost of product support and less about the need for end-user training, to some extent. Karat identifies the reduced cost of product support as a potential benefit but does not provide further discussion or examples of it. Karat does not identify the reduced cost of end-user training as a potential benefit (Table 6).

Mayhew and Mantei, on the other hand, provide sample calculations for these benefits. To calculate the benefits in product support, they use the number of customers, reduced number of calls per year per customer, the length per call, and the hourly wage of the customer support. To calculate the savings in end-user training, they use the number of customers, number of training classes per customer, the length of training per class, and the hourly wage of the trainer.

Table 6. The extent to which usability cost-benefit models identify and document benefits for customer support.

Benefit category	Ehrlich & Rohn	Karat	Bevan	Mayhew & Mantei
Reduced cost of product support	XX	X	XX	XXX
Reduced cost of end user training	X	-	XX	XXX

XXX = The benefit is identified and well documented, concrete guidelines, examples, etc.
XX = There is some discussion about the benefit, no concrete guidelines
X = The benefit is identified
- = The benefit is not identified

None of the analyzed models suggests different approaches for estimating the benefits for customer support in different cases: whether the product is a product tailored for a particular customer, or whether the product is mass produced as a shelf product. For example, a development organization may be more likely to provide customer support for users of a tailored product than when the product is sold in shrink wrap off the shelf. It can also be argued that estimating the benefits of better usability is somewhat different when the customer is internal in a development organization or when the support is part of the business of the development organization.

4.1.8 Benefits in daily use

Two categories of usability benefits are identified during a product's use: increased productivity and less need for end-user support. The end user can benefit from higher productivity when the most frequent tasks take less time. It is estimated that productivity within the service sector would increase 4-9% annually if every product were designed for usability (Landauer 1995). Productivity is increased when using more usable products through decreased task time, less rework, and greater work satisfaction.

All models identify increased productivity as one benefit. Karat provides a couple of examples of how to calculate it. Some savings can be made if there is less need for active product support in a development or customer organization. Ehrlich and Rohn, Bevan, and Mayhew and Mantei identify the lesser need for end-user support as a potential benefit (Table 7). According to Ehrlich and Rohn, a product that is not easily used or well explained can reduce profits by millions of dollars if the company has a low profit margin or a large customer base. Mayhew and Mantei have an example about calculating the increased productivity but they do not give a concrete guideline and there is little discussion about this benefit in general. Karat has some discussion about this benefit and a very brief guideline. Ehrlich and Rohn identify the increased productivity as a possible benefit, but there is no further discussion about it.

One possible benefit could be the indirect effect of better usability when its effect on a mission critical system reduces the problems of other systems using it. The analyzed models do not, however, identify this benefit.

Table 7. The extent to which usability cost-benefit models identify and document benefits for customers and end users.

Benefit category	Ehrlich & Rohn	Karat	Bevan	Mayhew & Mantei
Increased productivity	X	XX	XX	XX
Less need for end-user support	XX	-	X	XX

XXX = The benefit is identified and well documented, concrete guidelines, examples, etc.
XX = There is some discussion about the benefit, no concrete guidelines
X = The benefit is identified
- = The benefit is not identified

4.1.9 Costs

Two main categories of the usability costs in the development phase can be identified: one-time costs and sustaining costs. The sustaining costs include cost of usability activities and cost of redesigning the prototype. Ehrlich and Rohn have a detailed discussion about one-time costs and examples of sustaining cost, but the cost of prototype redesign is not identified. The sustaining cost of usability activities is identified in all models. Mayhew and Mantei have some examples of calculating the sustaining cost of usability activities, but there is no further discussion or guidelines about the calculations. Bevan mentions this benefit only briefly and makes reference to Bias and Mayhew for further information. The sustaining cost of prototype redesign is identified by Karat. Mayhew and Mantei also identify that benefit and have a simple example calculation (Table 8). None of the models identifies the costs after the development phase.

Table 8. The extent to which usability cost-benefit models identify and document the costs.

	Ehrlich & Rohn	Karat	Bevan	Mayhew & Mantei
One-time costs	XX	X	-	XX
Sustaining cost of usability activities	XX	X	X	XX
Sustaining cost of prototype redesign	-	X	-	XX

XXX = The benefit is identified and well documented, concrete guidelines, examples, etc.
XX = There is some discussion about the benefit, no concrete guidelines
X = The benefit is identified
- = The benefit is not identified

4.1.10 Summary of findings

Paper I contributes to the thesis by analyzing the different cost-benefit analysis models that are used for estimating the costs and benefits of better usability through usability activities. The analyzed models have a slightly different approach for identifying, categorizing, and assessing the benefits of usability. All the models addressed the increased sales of a more usable product as one of the benefits, but none of the models distinguishes between these benefits for business-to-business and business-to-consumer products. Only Ehrlich and Rohn and Bevan include increased customer satisfaction as a potential business benefit. From all the analyzed models, only Bevan identifies savings from a reduced cost of the future redesign of the architecture by fixing usability problems for future versions of the product. An easier tailoring of the product through user-centered design as a potential benefit is not explicitly discussed in any of the models.

All the analyzed models approach usability benefits through some sort of cost-benefit analysis. The identified benefits of better usability are measured against the estimated costs of usability activities. Every model analyzes the costs and benefits of using user-centered design and not the overall benefits of better usability of the product.

Analyzing the business benefits of better usability is not an easy task. Some of the potential benefits can be estimated quite easily. For example, the benefit of a lessened need for product support is rather straightforward to calculate. Some of the potential benefit areas are, however, quite abstract and it is therefore difficult to estimate those benefits. For example, it is very difficult to estimate what impact better usability has on improved company reputation, even when it is clear that poor usability hurts company reputation (Mauro 1994).

Some of the existing models also analyze the benefits of better usability from the end user's viewpoint. The potential benefits for end users are much more difficult to calculate than benefits for development or customer organizations. Also, the potential benefit areas for end users are harder to assess economically, even when there is a link between poor usability and higher rates of absenteeism, less job satisfaction, and increased turnover (Schneider 1985). Some of the analyzed models include increased work productivity as a benefit for end users. It can be argued that the benefits from increased productivity can be calculated more easily from the viewpoint of the customer organization.

In some existing usability cost-benefit models, the benefits are seen from the point of view of a starting development project. This approach does seem to be a bit problematic, because some of the potential benefits are clearly directed to a whole organization, and it may not be very useful to estimate those benefits from the point of view of a development project. For example, it is not very important to reduce support costs for a development project because they are not directly affected by the cost of product support. The models also have differences in regard to who does the usability benefit analysis and identifying the target group of the analysis (Rajanen 2002). When the potential usability benefits are analyzed from an organizational point of view and the business type of the development organization is identified as a necessary variable in usability cost-benefit analysis, all possible benefits can be fully taken into account.

4.2 Categorizing the usability cost-benefit analysis models (ECITE 2006)

Paper II is a conceptual-analytical study based on a literature review and contributes to the thesis by categorizing the usability cost-benefit analysis models based on the approach they take to the usability cost-benefit analysis and identifying the conductor of the analysis. This paper answers the following research questions:

1. How do the usability cost-benefit models approach the usability cost-benefit analysis?
4. What kind of empirical background do the models identify and document?
5. What interest groups do models identify for usability cost-benefit analysis?

Knowing how the usability cost-benefit model approaches the usability cost-benefit analysis helps the user of the model to identify a model that is the best suited approach for his/her context. The identified and documented empirical background of the model may help in choosing the model that is tested in a similar context. Also, a model with a clearly identified and documented empirical background may be seen as being on more solid ground than a model with little or no identified empirical background. Identified

interest groups for usability cost-benefit analysis may help to choose a model that is best suited to the intended organizational context of use.

Each of these analyzed models is compared against three categories to identify the different characteristics. These categories are based on the collection and categorization of usability cost-benefit models and reported usability benefit cases from the literature (Rajanen & Jokela 2004, Rajanen 2003). These three categories are:

1. The approach for usability cost-benefit analysis
6. Empirical background behind the usability cost-benefit model
7. Interest groups identified in the usability cost-benefit model

4.2.1 The approach of usability cost-benefit analysis

Two different categories of usability cost-benefit analysis were identified from the analyzed usability cost-benefit models: the user-centered design approach and usability task approach. In the user-centered design approach, the costs and benefits of better usability are seen through the overall user-centered design process, while the usability task approach has a narrower focus on the costs and benefits of individual usability tasks. From the analyzed usability cost-benefit analysis models, the models of Ehrlich and Rohn and Donahue have the user-centered design approach where the costs and benefits of user-centered design activities are identified. From the analyzed usability cost-benefit analysis models, the models of Ehrlich and Rohn, Bevan, Karat and Mayhew and Mantei have the usability task approach where the costs and benefits of individual usability tasks are identified. The models of Ehrlich and Rohn and Bevan also identify the need to analyze the costs and benefits of some of the individual usability tasks in addition to user-centered design costs and benefits. Since the user-centered design can be done with a wide variety of individual usability tasks, it can be argued that the user-centered design approach for usability cost-benefit analysis may be more comprehensive than the usability task approach. New usability cost-benefit models should, therefore, primarily have a user-centered design approach and may also identify some important usability tasks for cost-benefit analysis.

Table 9. The approach for usability cost-benefit analysis

	Ehrlich & Rohn	Bevan	Donahue	Karat	Mayhew & Mantei
Approach	UCD/UT	UCD/UT	UCD	UT	UT
UCD = User-centered design approach UT = Usability task approach					

4.2.2 The empirical background

User-centered design and usability activities are very down-to-earth and practical methods. Therefore, the usability cost-benefit analysis model should have a strong empirical background (Mauro 2002). The usability cost-benefit models should be based on empirical research in reported cases in the commercial SW development context, or on empirical research that has been used to verify the model. It can be argued that a model with a strong identified empirical background is more believable than a model with little or no identified empirical background. The models have differences in identifying the empirical background of the model. From the analyzed usability cost-

benefit analysis models, only Karat documents all of the empirical background of the model in detail, detailing the individual case studies behind the development and evaluation of the model. Ehrlich and Rohn and Mayhew and Mantei document some of the empirical background, but not in the same amount of detail as Karat, and, therefore, the identified empirical background is weaker in these models. Bevan and Donahue do not identify the empirical background behind these models. New usability cost-benefit models should have, identify, and document their strong empirical backgrounds fully.

Table 10. The empirical background behind the models

	Ehrlich & Rohn	Bevan	Donahue	Karat	Mayhew & Mantei
Identified empirical research	XX	-	-	XXX	XX
XXX = The model has a well-identified empirical background XX = The model has an identified empirical background X = The model has an empirical background but the empirical research is not identified further - = The model does not mention empirical background at all					

4.2.3 Interest groups

The usability cost-benefit analysis can have a different focus depending on the conductor and the target group of the analysis. Ehrlich and Rohn and Karat identify the usability team member as the conductor of the usability cost-benefit analysis. Other usability cost-benefit analysis models do not identify the conductor of the analysis at all. Ehrlich and Rohn, Karat, and Mayhew and Mantei identify organizational management as the target group of the usability cost-benefit analysis. Ehrlich and Rohn and Bevan identify project management as the target group of the usability cost-benefit analysis. Donahue identifies neither the conductor of the analysis nor the target group of the analysis. It can be argued that the best effect of introducing usability activities into a development project is achieved when the requirements for better usability are handed down by organizational management to a development project (Rajanen 2003). Therefore, it can be argued that the usability cost-benefit analysis has more impact and results if the target group of the analysis is the organizational management. New usability cost-benefit models should, therefore, identify the organizational management as the primary target group for the cost-benefit analysis.

Table 11. The identified interest groups for usability cost-benefit analysis

	Ehrlich & Rohn	Bevan	Donahue	Karat	Mayhew & Mantei
Conductor of analysis	U	-	-	U	-
Target group of the analysis	O, P	P	-	O	O
U = Usability team member O = Organizational management P = Project management - = The model does not identify the conductor or the target group					

4.2.4 Summary of findings

Paper II contributes to the thesis by categorizing the usability cost-benefit analyzing models based on the approach they have to the usability cost-benefit analysis and identifying the conductor of the analysis. The analyzed models have different approaches for identifying the costs and benefits of usability and identifying the interest groups of the usability cost-benefit analysis. The models also vary in identifying the empirical research on which the models are based. Therefore, it can be assumed that they are built upon a different basis and that they are made to fit different specific purposes (Rajanen 2002). The models also have differences in who does the usability cost-benefit analysis and what the target group of the analysis is.

In some of the usability cost-benefit models, the benefits are seen from the point of view of a starting development project and are identified by a member of the usability team. This approach might be seen as problematic because some of the potential benefits are clearly directed to an entire organization, and it may not be very useful or possible to estimate those benefits reliably from the point of view of a development project or usability team. For example, it is not very important to reduce support costs for a development project because they are not directly affected by the cost of product support, and the impact on support costs may be difficult to estimate at the development project level because they lack the necessary data from the organizational level. When the usability cost-benefit analysis is done from an organizational point of view and the business type of the development organization is identified as a necessary variable in usability cost-benefit analysis, all possible benefits can be fully taken into account.

The usability cost-benefit analysis models have two different approaches for identifying the costs and benefits of usability. Two of the models (Ehrlich & Rohn and Bevan) approach usability cost-benefit analysis through identifying the costs and benefits of user-centered design activities. Four of the models (Ehrlich & Rohn, Karat, Donahue, and Mayhew & Mantei) approach usability cost-benefit analysis through identifying the costs and benefits of individual usability tasks. The usability cost-benefit analysis models also vary in identifying the empirical research upon which the models are based. One of the models (Karat) identifies the empirical background of the model in detail. Two of the models (Ehrlich & Rohn and Mayhew & Mantei) identify some of the empirical background behind the models. Two of the models (Karat and Donahue) do not identify the empirical background behind the models.

The usability cost-benefit models also have differences in identifying the interest groups of the usability cost-benefit analysis. Two of the models (Ehrlich & Rohn and Karat) identify a member of the usability team as the conductor of the analysis. Three of the models (Bevan, Donahue, and Mayhew & Mantei) do not identify the conductor of the analysis at all. Three of the models (Ehrlich & Rohn, Karat, and Mayhew & Mantei) identify the organizational management as the target group of the usability cost-benefit analysis. Two of the models (Ehrlich & Rohn and Bevan) identify the project management as the target group of the usability cost-benefit analysis. One model (Donahue) does not identify the target group of the usability cost-benefit analysis.

5 Results: Usability cost-benefit considerations in commercial and OSS development contexts

5.1 Usability cost-benefit considerations in the commercial development context (Interact 2007)

Paper III contributes to the thesis by contrasting the usability cost-benefit analysis literature and the usability cost-benefit considerations with an empirical case in the commercial software development context, in which usability cost-benefit considerations, when addressed together with other usability activities, resulted in usability becoming a curse word. An interpretive case study (Klein & Myers 1999) was carried out in an SW development organization. In an extensive literature search, no similar interpretive case studies were found where usability cost-benefit analysis models would have been used in an empirical setting and the results would have been contrasted with the usability cost-benefit analysis literature. Empirical analysis reveals that clearly divergent meanings and motives have been attached to usability and its cost-benefit analysis in the organization, and these resulted in failure of the intended introduction of usability activities into this organization and eventually the organization abandoning usability activities altogether.

A detailed analysis of the literature offering advice on usability cost-benefit analysis was carried out. Based on the literature review, an analytical framework for the empirical analysis has been developed and utilized in making sense of the empirical data derived from one case organization, into which usability cost-benefit analysis was introduced along with other usability activities. The existing literature also highlighted that very divergent meanings can be attached to usability in practice, the studies showing that it has been used only as a buzzword or as a mere slogan without any proper understanding of it (Artman 2002, Catarci *et al.* 2002, Iivari 2006, Tudor 1998). Due to the assumed importance of the usability cost-benefit analysis, but also keeping in mind the risk that this term and its analyses may be used and interpreted in a multitude of different (and even conflicting) ways, this case study examined the process of meanings negotiation related to usability and its cost-benefit analysis in one case organization.

5.1.1 Analytical framework

Table 12 summarizes the costs and benefits of usability outlined by the usability cost-benefit analysis models presented above. The focus in the empirical analysis was on the development context because the empirical data have been gathered solely from that context. However, the benefits to be achieved through better usability in the use context (including both the customers making the buying decisions and the end users) are also summarized in Table 12. This is because it was assumed that in the development context, one should acknowledge the benefits achievable in the use context while motivating the usability activities in the development context.

Table 12. Analytic framework for empirical analysis of usability costs and benefits

	Development context	**Use context**
Benefits	Increased sales	Reduced training time
	Reduced support costs	Increased productivity
	Reduced development costs	Increased (customer, user) satisfaction
	Reduced training costs	Reduced staff turnover

Costs	One-time costs	-
	Recurring costs	
	Redesign costs	

In addition, the benefits can be separated into tangible and intangible benefits (Karat 1994). However, advice for the cost-benefit analysis is offered only in relation to the tangible benefits. Furthermore, a noteworthy observation is that the models mostly assume that the benefits can and should be quantifiable. In sum, the empirical analysis was based on the assumed usability costs and benefits listed in Table 12, also acknowledging the distinctions between the tangible/intangible and quantifiable/non-quantifiable benefits mentioned above.

5.1.2 Summary of findings

Paper III contributes to the thesis by contrasting the usability cost-benefit analysis literature and the usability cost-benefit considerations with an empirical case in the commercial software development context. The empirical analysis revealed that very divergent and surprising meanings and motives were attached to usability and its cost-benefit analysis in the case organization. Increased sales and reduced development costs were strongly emphasized as benefits of better usability. However, very surprising meanings were attached to them both. Reduced training and support costs, on the other hand, were only mentioned. From the viewpoint of the use context, increased customer satisfaction was acknowledged. However, the increased development costs associated with better usability were the main failure factor of the whole usability improvement effort. Results from similar studies show that even if managers maintain that usability expertise is crucially important for usability work, there is limited willingness to actually increase the number of usability staff and to allocate resources to usability work, therefore creating a contradiction within the organization (c.f. Cajander *et al.* 2006).

Table 13. Empirical results in relation to the analytical framework on usability cost-benefits

Usability Cost-Benefits	Empirical findings
Benefits for the development context	*- Increased sales through use of usability as a tool in sales and marketing, as a tool for convincing the customer, resulting in marketing demos that sell themselves and conquer the world*
	- Reduced development costs through usability keeping the customers out of the development
Benefits for the use context	*- Increased customer satisfaction that, however, needs to have a positive, visible effect on company's profit*
Costs for the development context	*- Increased (recurring and redesign-related) development costs (actually UI design costs that would have been realized in any case), usability activities condemned as ineffective, labor intensive and time-consuming*

Next, the implications of the empirical results are discussed in relation to the existing usability cost-benefit analysis literature. First of all, it must be emphasized that usability cost-benefit analysis models need to recognize more clearly that it will take time for the usability benefits to be realized, and the costs will be evident much earlier. In all, time is

an important issue for the development projects. The project managers may be hesitant to introduce any usability activities to their project because they fear that those activities will only consume more time, and the promised savings in the development time through less need for redesign are quite vague. The costs of usability activities are very much tangible and quantifiable, but the possible benefits are quite intangible and usually not easy to quantify reliably. Also, the cost of better usability is to be paid early in the development project, whereas the promised benefits of better usability may or may not be achieved in the distant future.

Altogether, the usability cost-benefit analysis models seem to highlight issues that are either too insignificant, too vague, or solely aim to serve business needs, neglecting the interests of the end user. In this empirical case, it can be argued that the usability cost-benefit models did not succeed in raising the right issues to convince the management to continue the usability improvement effort. The potential benefits of usability already acknowledged in the case organization seemed to be insufficient for the management. In addition, some of the possible benefits of better usability that the usability cost-benefit models identify were considered insignificant. As mentioned, reduced training and support costs were ignored in the case organization. In addition, the managers pointed out that even more insignificant issues are highlighted in these models. For example, in one of the project workshops, after a researcher presentation arguing that savings in printing costs of product manuals should also be counted as a benefit from better usability, a manager from another participating company made a very critical comment that "This is peanuts. We should not spend any time discussing this issue. When I am handling projects costing several millions, why should I care about saving a few hundred through having to print fewer pages for the product manual?" (Field notes). Having to print fewer pages for the product manual was raised as a possible benefit that could be gained through usability activities and, therefore, result in better usability by many usability cost-benefit analysis models; but the managers in the workshop considered calculating and even discussing this benefit to be simply a waste of time.

Regarding neglecting the interests of the user, this case shows that the management, sales, and development were only interested in the paying customers and did not show interest in the end users. It can be argued that the usability cost-benefit models did not provide the right benefits, so the management, sales, and development would have paid more interest to the users. One could even argue that the usability cost-benefit considerations might have acted not as a wake-up call (cf. Schaffer 2004) in the intended way, but as a negative wake-up call by directing the attention solely to the paying customer and to the finances of the development organization, which, of course, also need to be considered, but which should not result in the total neglect of the end-user interests. Using usability cost-benefit considerations as motivation for improving usability activities resulted in a total failure in this particular case study.

It can be argued that the results achieved in Paper III are very alarming from the viewpoint of socially responsible and morally sound HCI. The management-identified goals such as 'taming the customer' and 'improving the image of the company' by appealing to usability can be criticized as being a morally ambiguous 'misuse of usability,' since, in this situation, one does not necessarily develop usability at all, but only uses it to convince and, in the worst case, to hoax the customer. The management goal of 'taming the customers' with the help of usability might even be viewed as a way of 'silencing the users,' instead of 'giving them a voice' (Asaro 2000). This can also be interpreted as a form of technological colonialism (Asaro 2000) only dressed in the gown of 'usability.' In this case, it can be argued that the users and customers are being

'colonized' by appealing to usability for the sake of the management goals of the development organization. It can be argued that this kind of 'misuse of usability' runs against the noble principles of the usability specialists and the HCI tradition, where the purpose is to understand and appreciate the end users in particular and to provide design solutions with good usability to serve them in the best way possible. Not surprisingly, studies have shown that management perceives orderliness, objectivity, and control as key success factors, and from that perspective, usability can be seen as unclear and difficult to incorporate in the business (Cajander *et al.* 2006).

Altogether, according to Spinuzzi, this type of financially driven orientation can be viewed as the realization of Scandinavians' worst fears (Spinuzzi 2002) referring to the Scandinavian IS research tradition (e.g., Greenbaum & Kyng 1991) that has advocated workplace democracy and union involvement in the development of computer systems. The tradition relied on the notion of conflict between management and labor, and positioned itself strongly on the side of the labor against the 'oppressing' management. One could argue that usability people and usability cost-benefit analysis models should be positioned on the side of 'the user,' not 'the manager'; i.e., they should aim to ensure that the usability efforts are beneficial, especially for the user, even though hopefully also for the other stakeholder groups. In all, regarding research on usability cost-benefit analysis, it can be argued that the researchers should carefully consider the different kinds of interpretations of usability costs and benefits, and particularly the different kinds of uses of their analysis revealed in this paper. In addition, it can be argued that the research community should take some responsibility for these uses and interpretations, or at least consider how to advocate more 'appropriate' uses and interpretations.

5.2 UKKOSS cases (JITTA 2011)

Paper IV contributes to the thesis by exploring the OSS development context in general and specifically the difficulties and intricacies of introducing usability activities into OSS development projects. The case projects involved in this study were OSS development projects that were sufficiently large to ensure that these projects would remain active during the period of usability team involvement. OSS development projects with just one or very few core developers can quite easily become inactive for uncertain periods of time, when one of the main core developers is not active for some reason. Three of the projects were also small enough to allow for easy identification of the hierarchical structure of the project and, therefore, easy communication with the core developers. These OSS development projects also developed the software for non-technical end users. The third case project was much larger and involved the development of software for highly skilled users. The case projects were deliberately chosen from projects that did not involve software companies. This was because OSS development projects with company involvement may have usability resources that they utilize during development (e.g., Benson *et al.* 2004, Frishberg *et al.* 2002, Iivari *et al.* 2008, Nichols & Twidale 2006). The OSS projects that have been started and/or closely monitored by software companies might not be very different from proprietary software development from the usability activities point of view. Therefore, these were not selected as cases and focus was solely on the OSS development context proper.

5.2.1 Partnership with the OSS community

The HCI literature recommends that when introducing usability activities into software development, one should gain a thorough understanding of the context into which the activities are to be introduced in order to select the most suitable approach, since there is no 'one size fits all' (Aucella 1997, Iivari 2006). In the OSS context, the argument has been made that usability activities should be tailored to fit the OSS development philosophy and culture (Benson *et al.* 2004, Bødker *et al.* 2007, Cetin *et al.* 2007, Nichols & Twidale 2003, Terry *et al.* 2010, Zhao & Deek 2005). It has also been suggested that the developers are the most important target group and that they should perceive usability specialists as allies (Aucella 1997, Bloomer & Croft 1997, Fellenz 1997, Mayhew 1999, Rosenbaum *et al.* 2000, Schaffer 2004). The point has also been raised that usability specialists are needed in OSS development, but their position tends to be quite challenging in OSS projects (Andreasen *et al.* 2006, Bach *et al.* 2009, Cetin *et al.* 2007, Nichols & Twidale 2003, Twidale & Nichols 2005).

These four empirical cases demonstrated that the usability specialist should not and cannot be an outsider in OSS development, but should integrate with and become an active, visible member of the community. All four of the cases offer us evidence related to this, but UKKOSS 2 offers the most interesting evidence. Although the importance of the leading core developer having a personal interest and approval for usability activities in the second case was evident, other members of the core development team clearly did not want to be left out and wanted their voices to be heard as well. In UKKOSS 2, the OSS project was clearly interested in users and in improving the usability of their software, but a lack of knowledge also apparently existed regarding usability and its potential benefits. This supports the argument that the status of OSS usability is years behind the status of usability in commercial software development, as reported in the literature (cf. 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 second case project also showed that the usability activities performed by the usability team were a successful wake-up call for the developers (cf. Schaffer 2004) and interest in usability activities continued long after the usability team had finished its work. In UKKOSS 3, many core developers and community members had strong opinions about changing the user interface. Any suggestions regarding changing the user interface to even slightly resemble any of the competing commercial software interfaces resulted in bitter resistance and arguments, even if these changes would improve the usability. Consequently, this case implies that the community as a whole should be committed to usability. In this case, the usability team did not succeed in becoming recognized members of the community, even though lots of effort was devoted to this. One core developer published news related to the work of the usability team; however, due to some misfortunes, this did not happen at the most critical time and was, therefore, not adequate.

Based on the results of these four cases, it can be argued that the usability specialists should 'infiltrate' the development community, make themselves visible by providing information about usability and by evaluating and producing design solutions (cf. ISO 13407 1999), and offer their usability expertise to the core developers. These findings are in line with the findings by Bach and colleagues, who also identify educating the OSS community and building trust and community as possible ways of improving the position of usability or user experience (UX) design in OSS development (Bach & Carroll 2009, Back *et al.* 2009). Although the core developers may have a limited understanding of usability, the usability specialist should approach the core developers and contributors as

peers rather than as pupils. Dialogue between the usability team and the development team is vital for the introduction of usability activities and for the usability activities to have any effect on development. One should be able to adapt the usability activities to the development phase, culture, philosophy, and the vision of the core developers and the community. This conclusion is connected to the findings from cases 2 and 3 related to the rapid development of the OSS, and to the difficulty of targeting the usability activities to the right version at the right time. The usability activities need to be adapted to the way in which the rest of the development works, but at the same time, a certain distance also needs to be kept in order to maintain an objective view and to keep the interests of the non-technical end users in mind. This was brought up in the UKKOSS 2 case, where the developers thought that the usability team should keep its distance from them in order to maintain an objective view, but at the same time become a close-knit part of the development team, adapting to its culture and ways of working. The contradiction is apparent and raises very interesting challenges for adapting usability activities to OSS development. In OSS development, as well as in the commercial development context, the challenge is for the usability specialists to align their ways of working with the engineers (Mayhew 1999) but, at the same time, preserve their role as the 'representatives of non-technical end users' (cf. Iivari 2006, Rajanen & Iivari 2007).

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

5.2.2 Management commitment in the OSS development context

Another finding from the traditional usability literature is the importance of management commitment and support when introducing usability into software development (Fellenz 1997, Schaffer 2004). Cost and benefit tradeoffs need to be considered, and the management may be concerned with the possibility of usability activities increasing the development costs and time needed (Bloomer & Croft 1997, Vredenburg *et al.* 2002). This discussion bears interesting implications for the OSS development context, as, to a certain extent, the core developers can be argued to have the same characteristics as

managers in commercial software development. The core developers have to make both low-level decisions regarding whether or not a particular contribution is accepted by the code, mid-level decisions regarding software features to be included in individual releases, and strategic decisions regarding the direction of future development.

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

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

UKKOSS cases 2 and 3 offer some evidence related to management commitment in OSS projects. Although, in the second case, most of the feedback was positive, not all of the suggestions made by the usability team were accepted. The core developers essentially considered these findings and suggestions and picked the ones on which they were able to agree (in line with, e.g., Ye & Kishida 2003). If the developers did not think that a certain issue was a problem, they would just ignore it and not include it in their "to do" list or development roadmap. This can be compared to the way in which patches are committed; if the core developers like the patch, it will get into the official release (cf. Aberdour 2007). If they do not like the patch, they can easily ignore it and it will not get into the official release. Similarly, the core developers can easily ignore usability activities if they do not know very much about usability in the first place. This was also apparent in the third case, in which it was difficult for the usability activities to get noticed without explicit core developer support. Without a certain critical mass of usability advocates and experts, it can be difficult to establish the legitimacy of usability arguments against countervailing expert-user functionality-centric claims (Twidale & Nichols 2005).

Interestingly, many different potential benefits of better usability were brought up in the discussions between core developers and community in the third case project (cf. Benson *et al.* 2004). As mentioned in the literature, it is important to be able to show the usability benefits that can be achieved (Mayhew 1999, Rosenbaum *et al.* 2000). These kinds of considerations are especially important for management (Bloomer & Croft 1997, Bias & Mayhew 1994), and also likely for the core developers in the OSS development

context as they share similar positions and tasks like the management in the commercial SW development context. Indeed, some potential usability benefits were identified in the third case, including, for example, those related to gaining more users for the software if it would be made easier to learn, and having talented 3D content creators transfer from using similar commercial products if transfer from commercial 3D content creator software would be made easier by allowing the users to customize some key parts of the user interface and shortcuts to their liking. This potential usability benefit can be seen as being similar to increased sales in the commercial SW development context (c.f. Mayhew & Mantei 1994, Karat 1994, Ehrlich & Rohn 1994, Bevan 2000, Donahue 2001). However, these identified usability benefits were not sufficient to result in a wake-up call (cf. Schaffer 2004) for the need for usability activities among the core developers.

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

5.2.3 Summary of findings

Paper IV contributes to the thesis by exploring the OSS development context in general and specifically the difficulties and intricacies of introducing usability activities into OSS development projects. The main findings of the four empirical studies can be summarized as follows. To bring usability activities successfully into the OSS project, the usability specialists should adopt a participative instead of a consultative role, and:

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

Many highly practical findings were identified in Paper IV for people interested in introducing usability activities into OSS projects.

5.3 Fitting usability cost-benefit consideration into the OSS context (ECIS 2010)

Paper V contributes to the thesis by further exploring the usability costs and benefits in the OSS development context. The status of usability in OSS development today resembles the status of usability in commercial software development years ago when there were little or no usability specialists and no usability activities integrated into the

development process (cf. Andreasen *et al.* 2006, Benson *et al.* 2004, Cetin *et al.* 2007, Nichols & Twidale 2003, Nichols & Twidale 2006, Zhao & Deek 2005, Zhao & Deek 2006). Therefore, it can be assumed that the measures used for advocating usability in commercial software development could now be of use in OSS development, but with some modifications to take into account the different nature of the OSS development context. Paper V explores how the potential usability costs and benefits from traditional usability cost-benefit analysis literature should be refined to fit into the OSS development context.

In Paper V, 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 & Myers 1999) addressing these two OSS development models have been carried out. The results of the case studies are 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 with no deep technical knowledge. Therefore, the usability of OSS has become very important; 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 results of two case studies are discussed in this paper. 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 utilized research methods related to them are briefly outlined next. More information is found in Iivari (2008), Iivari (2009), and Iivari *et al.* (2008).

In this analysis, all the data gathered were carefully read through and relevant pieces of information were extracted and categorized as ‘benefits’ or ‘costs’ of usability or as having implications for the ‘benefits’ or ‘costs’ of usability as outlined in the existing models. The analysis was data driven. The existing OSS literature was also used to supplement the empirical data derived from these two cases.

Based on both the existing OSS literature and the findings from these cases, it can be claimed that the usability cost-benefit models need to be altered somewhat to fit the OSS development context. Before modifying them, the case findings related to the implications of the OSS for the existing usability cost-benefit analysis categories are presented.

5.3.1 Implications of the OSS for usability benefits

The increased sales were identifiable as a potential usability benefit category from both case studies, but in a somewhat different sense than traditionally. In the community OSS development, the developers want to attract more users and keep their existing users happy. A large user base is an important source of status, prestige, and peer recognition, even though there is nothing sold to the users like the usability cost-benefit models assume. Also, as a project becomes more successful, it will attract more competent developers who wish to get their share of the glory and set some of the attention (Aberdour 2007). In company OSS development, user satisfaction and good UI are

important image and competitive factors for the company. These can have an impact on sales and they are in line with how usability cost-benefit models present this benefit (cf. Rajanen 2006, Rajanen & Iivari 2007).

The OSS development context also has certain implications for the category related to the reduced development costs. Early and continuous releases are characteristic of OSS development and they enable early feedback gathering and redesign. In community OSS development, the community actively comments on and redesigns the solution during that development phase, thereby 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 implement, which is in line with how usability cost-benefit models outline this benefit (cf. Rajanen 2006, Rajanen & Iivari 2007).

Table 14. Implications of the OSS for usability benefits

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

5.3.2 Implications of the OSS for existing usability costs

The implications for the category related to reduced training and support costs can also be identified from the OSS development context. In the OSS development context, user support is provided as peer support in the OSS forums. In the case of company involvement, the OSS community might also provide user support in the forums, which can be considered a benefit, if the company is not interested in making money through providing user support by itself. The implications of the OSS for the existing usability benefit categories are summarized in Table 14.

The OSS development context also has certain implications for the costs outlined in the usability cost-benefit analysis models. One-time costs can naturally be identified from company OSS development where a 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. The 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 the company OSS development context within which usability specialists are hired, but this resembles the situation in commercial 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. Redesign costs can, however, easily be identified both in the community and the company OSS development contexts. In the company OSS development context, more time and opportunities for redesign probably increase the redesign costs when compared to the closed source software development. In the

community OSS development context, 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 can clearly be associated with OSS development. The implications of the OSS for the existing usability cost-benefit categories are summarized in Table 15.

Table 15. Implications of the OSS for usability costs

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 are 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

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.

5.3.3 Usability benefits in the open source software development context

Regarding the benefits of usability, it can be argued that increased sales should be perceived as increased popularity or distribution of software. In the OSS context, users do not pay for the software, but a large user base might be a source of reward, prestige, 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 are, 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 the company OSS development contexts. Usability cost-benefits models approach the reduced development costs through less need for development resources and earlier entry into the market (Rajanen 2006, Rajanen & Iivari 2007). In both the community and the company OSS development contexts, the community can take an active part in redesign and, therefore, reduce the development time, even if the development project does not have any salaried positions and costs in a similar sense as is the case in the company context. The OSS development is very iterative and rapid by nature (Raymond 1999), so early release to the community for feedback gives more time for iteration and redesign. Developers in both the community and the company OSS development contexts can save time when there is less pressure for redesign through better usability that is already achieved. 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 the early phase through usability activities. Also, the non-technical users can contribute to the project in a more manageable manner though usability methods.

As mentioned earlier, company OSS development can use the community as a provider of training and support. If providing support is an important part of the 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, thereby reducing the support-based earnings in any case (cf. Rajanen & Iivari 2007). The community OSS development context may 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 of both community and company OSS development. However, user support is a more mundane task that is not so much linked to reputation or other motivational factors affecting the will to contribute (Bergquist & Ljungberg 2001). User support is mainly performed by some end users who voluntarily answer the questions of other users (Singh *et al.* 2006, Lakhani & von Hippel 2000). This is clearly different from the categorization of this in the usability cost-benefit models, where peer support is seen as work that slows hidden cost in the use context (cf. Ehrlich et al 1994, Rajanen 2006), although studies indicate that techniques from knowledge management are needed to organize and make solutions sufficiently accessible to the help seekers so that they bother to try looking if their problem is already addressed (Singh *et al.* 2006). The refined usability benefits fitting the OSS development context are summarized in Table 16.

Table 16. Usability benefits in the open source software development context

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

5.3.4 Usability costs in open source software development context

While the benefits of better usability can be used as a motivator for introducing usability activities in the OSS development 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 also be identified in the community OSS development context in the sense that there needs to be a usability infrastructure, such as established discussion forums and usability guidelines for developers, but no physical infrastructure is needed since OSS development is typically distributed and without any form of

physical infrastructure. The company OSS development context is similar to closed source development. A usability infrastructure, e.g., a usability laboratory, may already exist or might need to be established, even though it is not mandatory. The company may also have one-time costs in establishing internet-based usability and OSS-related infrastructure such as forums or OSS repositories.

In company OSS development, the recurring costs are very much similar to commercial 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 on the web, but for some reason, these efforts have not yet succeeded 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 constantly requested 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 17.

Table 17. Usability costs in the open source software development context

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

5.3.5 Summary of findings

The existence of developer-users in the 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 the developer-user have an impact both on the development context and on 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 the OSS development context, the focus has traditionally been on cutting-edge technology and functionality, not on usability. OSS users have traditionally tolerated bad usability if highly innovative technological solutions have been provided. However, for the non-developer users, the usability may become a highly important factor, maybe even hindering the use of an otherwise suitable OSS solution. If OSS projects want to attract this kind of user, they need to start to emphasize usability.

The introduction, best methods, and possible benefits of usability in the 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.

Paper V contributes to the thesis by exploring the usability costs and benefits in the OSS development context. The analysis indicates that there were interesting parallels and differences when considering the costs and benefits of usability in the OSS development context. This exploration of the OSS development context and the analysis necessitated introducing a refined categorization and interpretation of usability costs and benefits with further implications both for theory and practice. The community OSS development context especially necessitates considerable modifications to the existing models since this environment has fundamentally different conceptions of what a ‘cost’ and a ‘benefit’ are.

6 Discussion and conclusions

"Whether your company does usability testing or not, your customers will, in effect, usability-test your system. Ultimately, relying on such 'usability testing by default' risks angering customers" (Donahue 2001).

The purpose of this dissertation was to identify and explore *if usability cost-benefit analysis is helpful when applied in commercial and open source software development contexts*. This broad topic was approached through three research questions that gave the following results based on the research.

6.1 Answers to the research questions

RQ1: What are the differences and commonalities of the existing usability cost-benefit analysis models?

The results from Papers I and II indicate that there are considerable variations in the usability cost-benefit analysis literature in how they identify and document the different categories of usability costs and benefits in different product development phases. None of the analyzed models fully documented all of the three steps of the cost-benefit analysis method as identified by Burrill and Ellsworth (1980) and most did not identify and document all the variables to be taken into account when making the investment decision based on the costs and benefits. The usability cost-benefit analysis models have variations in their approach and the identified empirical background. Some of the models do not identify any empirical background behind the model or identify the relevant interest groups for the usability cost-benefit analysis, such as the conductor and target group of the analysis. Most of the models identify the organizational management as the target group of the analysis, while some identify project management as the target of the analysis.

In order to build better usability cost-benefit models in general, the researchers should fully document all of the steps of cost-benefit analysis, identify and document all the variables to be taken into account when making the investment decision based on the usability costs and benefits, identify the empirical background behind their model, and identify the relevant interest groups of the analysis. Furthermore, in the OSS development context in particular, the usability cost-benefit analysis model should take into account the universal OSS development philosophy and principles in general and the hierarchical structure, community structure, and characteristics of a certain type of OSS development project in particular.

RQ2: How do the existing usability cost-benefit considerations fit into practice in the commercial development context?

The results from Paper III indicate that there are considerable risks of potential failure when using usability cost-benefit considerations in the commercial development context when the inherent costs of usability activities become apparent, concrete, and measurable while the potential benefits of better usability remain vague, uncertain, and unconvincing to the management. In the worst-case scenario, this lack of confidence in usability benefits may result in the management abandoning all usability activities like in the case described in Paper III. Also, the usability cost-benefit considerations included benefits that managers considered to be meaningless and a waste of time to even think about, such as the potential benefit of having to print fewer pages for the production manual through better usability. Also, some of the usability cost-benefit considerations

changed the role of usability in the eyes of the management from helping end users by better usability to usability as a publicity gimmick, a marketing slogan, and yet another development aspect that could and should be a potential target for cost cuts. The value of better usability as an ethical value itself disappeared behind financial considerations. Therefore, it can be argued that the usability cost-benefit considerations may be an uncertain and potentially very risky method of selling usability to project and organizational management and great care should be taken if it is used. In order to build better usability cost-benefit models and to minimize the possibilities of the model having unforeseen inherent risks as outlined in Paper III, the researchers should focus heavily on bringing forward the possible benefits of usability and not focus on those costs of designing and implementing user interface that would be realized in any case regardless of the usability activities.

RQ3: How do usability costs and benefits fit into the open source software development context?

The results from Papers IV and V indicate that it is possible to fit the usability cost-benefit considerations into the OSS development context even though there are no apparent financial or resource factors to be considered. In order to fit usability cost-benefit considerations into the OSS development context, the philosophy, principles, hierarchical structure, community structure, and characteristics of OSS development must be taken into account. For example, the increased sales as a benefit from better usability in the commercial SW development context turns into an increased user base and the inherent status that a large user base brings to developers in the OSS development context. The position of core developers in the OSS development context is, to some extent, similar to the position of managers in commercial SW development. Therefore, modified usability costs and benefits can be used as an argument for introducing usability activities to the OSS development context in a similar fashion to the commercial SW development context when the OSS philosophies and development context are taken into account in the usability cost-benefit categories. It is not easy to introduce usability activities into OSS development projects due to the fact that the usability specialists have to find these OSS development projects first and convince the core developers to integrate usability activities into the project roadmap. Usability cost-benefit considerations that fit into the OSS development context might help to convince the core developers that usability activities are important and should be integrated into the project roadmap.

6.2 Theoretical contributions

The theoretical contributions of the thesis can be divided into three parts. First, because there are no systematic studies about different usability cost-benefit analysis models, this research identified the strengths, weaknesses, differences, and commonalities between usability cost-benefit analysis models. The results from this part of the research indicate that there are considerable variations in the usability cost-benefit analysis literature in identifying and documenting the individual usability cost and benefit categories and the necessary steps of the outlined cost-benefit analysis in general. Furthermore, the usability cost-benefit literature has variations in its approach, empirical background, identified conductor of the analysis, and the intended target group of the analysis. Papers I and II introduce new criteria for evaluating and comparing usability cost-benefit analysis models.

Second, this research studied how the existing usability cost-benefit considerations fit into practice in the commercial development context and the results from this part of the research indicate that identified usability costs are far more apparent, believable, and measurable than the possible usability benefits. Therefore, there is a risk that the intended purpose of the usability cost-benefit analysis as an encouragement for usability improvement is jeopardized because the inherent costs of better usability become too apparent and scare the management off because they have an instrumental view of work for which efficiency and economy are important constituents (c.f. Cajander *et al.* 2006). Also, the literature introduces some usability benefits that have so little impact that the management may consider them meaningless to even consider—such as the identified benefit when printing fewer pages for product manuals—and, therefore, care should be taken when selecting the usability benefits as an argument for introducing the usability activities. Paper III provides a reported empirical case of using usability cost-benefit analysis in the commercial development context and raises new and unforeseen issues regarding fitting usability cost-benefit analysis into practice. No similar interpretive case studies were found where the results of using the usability cost-benefit considerations in the commercial SW development context would have been contrasted with the usability cost-benefit analysis literature.

Third, this research explored modifying the usability costs and benefits to fit them into the OSS development context. The results from this part of the research indicate that it is possible to fit the usability costs and benefits into the OSS development context even though the OSS development context lacks the financial and resource constraints that are the main reason for using the usability cost-benefit considerations in the first place in the commercial SW development context. The positions of core developers in the OSS development context and the managers in the commercial SW development context are, to some extent, similar and, therefore, usability cost-benefit considerations can also be used in the OSS development context when the OSS philosophies and development context are taken into account. Papers IV and V explore issues of bringing usability into the OSS development context, differences and commonalities between company and OSS development contexts, and possible usability cost-benefit considerations to be used in the OSS development context.

6.3 Practical contributions

This thesis has practical contributions in three areas. First, the results from Papers I and II can be utilized by usability cost-benefit analysis researchers as systematic criteria to further develop better usability cost-benefit analysis models. Also, the results from these papers can be utilized by company practitioners (e.g., managers, usability specialists, and developers) in choosing the most suitable usability cost-benefit analysis model or appropriate usability cost-benefit considerations for their development context.

Second, the results from Paper III can be useful for usability cost-benefit analysis researchers and company practitioners (e.g., managers and usability specialists) by identifying the potentially unforeseen risks of using usability cost-benefit considerations as an argument for usability activities in the commercial development context. The usability cost-benefit analysis literature does not identify these potential risks that, in the worst-case scenario, can result in the company abandoning the usability activities. There are no similar interpretive case studies reported in the literature where the usability cost-benefit analysis models would have been used in an empirical setting and the results would have been contrasted with the usability cost-benefit analysis literature.

Third, the results from Papers IV and V can be utilized by usability specialists to better bring usability activities into OSS development projects through a participative role, understanding the philosophy, principles, hierarchical structure, community structure, and characteristics of OSS development while using identified benefits of better usability as an argument. Also HCI and OSS researchers can utilize the results of Paper IV regarding using student teams in the OSS development context for gathering data and infiltrating the OSS development projects for research purposes. Developers, usability specialists, and community members can utilize the results from Paper V for advocating bringing usability activities into the OSS development project by identifying the possible benefits of better usability in the OSS development context.

6.4 Limitations and future work

Here, some of the limitations regarding the papers and this thesis as a whole are highlighted and further addressed.

In Chapter 4, this study has limitations related to the selection of the usability cost-benefit analysis models. Not all potential usability cost-benefit analysis models were explored in this study and those included in this study could have been further explored empirically by applying them individually in cases in the commercial development context in order to assess their suitability in practice. Furthermore, the usability cost aspect of usability cost-benefit analysis was not explored further though it has been explored in another paper (Rajanen 2007).

In Chapter 5, this study has limitations related to student involvement in the research process, having only one case for exploring usability cost-benefit considerations in the commercial development context and the lack of full-scale empirical testing of the usability cost-benefit considerations in the OSS development context.

First, the student involvement in the research process raises questions regarding the expertise of the students in conducting research and gathering research data in UKKOSS projects. The usability teams in the UKKOSS 1, 2, 3, and 4 cases consisted of students, but it can be argued that in this research, the student involvement was not as big a problem as it might have been because the author planned and closely supervised the work of the student projects, and students from this field are typically involved in OSS development projects anyway, as developers and community members. Nevertheless, student involvement can be seen as a limitation of this study that has to be taken into account when utilizing the results. However, it can be argued that the results would have remained the same even if professional usability people had tried the same consultancy-style approach, because the OSS developer culture places more value on the functionality of code than on interaction design (Green *et al.* 2009). Also, applying usability evaluation and UI design is best done prior to the start of development (Nichols *et al.* 2001, Nichols & Twidale 2003). Furthermore, OSS projects tend to be developed piece by piece by separate individual contributors, and comprehensive design and evaluation efforts do not fit the development process well (Green *et al.* 2009).

Second, the usability cost-benefit considerations were explored in only one case company in the commercial development context. However, by focusing on only one case, it was possible to analyze this particular case and the meanings, interpretations, and opinions about usability and its costs and benefits emerging inside the organization in greater detail than in multiple cases. This was due to the difficulties of getting access to companies and being able to conduct and publish research related to financial aspects of commercial SW development. Clearly, more empirical research employing a larger

amount of cases is needed in order to understand this phenomenon in depth. The data from the one case presented in this thesis were collected related in the larger usability improvement effort context and gaining access to a company for usability cost-benefit analysis research proved to be difficult. However, this one case proved to be invaluable by showing the worst-case scenario of what could happen when usability cost-benefit considerations really backfire and destroy all usability work within an organization. In addition, future research needs to be related to usability-cost benefit analysis, particularly when carried out in a professional manner, which was not the case in this case organization.

Third, while this thesis presents usability cost-benefit considerations that fit into the OSS development context, these presented usability cost-benefit considerations are not fully empirically tested in the OSS development project. An empirical test for these usability cost-benefit considerations in the OSS development context would require access to the core developer layer in several OSS development projects. These projects would have to have different sizes, hierarchical structures, and SW types. This empirical test would have to last a long time and would potentially require usability specialists to infiltrate the OSS development project and educate the core developers and community about basic usability methods and principles.

Fourth, the usability teams in the UKKOSS 1, 2, 3, and 4 cases were not involved with the case projects from the beginning of these projects. This was due to the long time that was necessary for an OSS project to evolve into a thriving community-supported development project. The usability literature argues that usability specialists should take part from the beginning of the development process. To be able to take part from the beginning of an OSS development project, the usability specialist should already have an established role in task and interface design when the project begins, and no software should yet have been produced. This would require the OSS developer initiating the project to have substantial knowledge and appreciation of usability to begin with. The OSS development philosophy of the "perpetual beta," where the software is constantly under redevelopment, is also a challenge for traditional usability activities and processes. As already mentioned, the software version that the usability team was testing in the second and third case was already old when evaluations were just beginning and, therefore, the usability team was constantly trying to shoot a moving target with its expert evaluations and improvement suggestions.

With regard to future areas of research, it would be interesting to determine how the effectiveness of the usability work would change if the specialists were even more integrated into the community. The potential risk of usability specialists losing their objective view as a result of this close integration with development when acting more in a participative than a consultative role should also be examined. Determining the effect of the size of the project on the easiness and acceptance of the usability activities would also be interesting. Another path for future work would be to study whether 'infiltration' with usability activities is more difficult to achieve within a tight OSS community hierarchy than it would be in an OSS community with a less rigid structure.. Interesting similarities and differences are also evident between OSS project leaders and managers of commercial software development projects that could be studied further. Further research into the possibilities of using the potential benefits of better usability as an argument for bringing usability activities into OSS projects would also be interesting. The methods, the level, and the difficulties raising usability in OSS development from grass roots activity into more organized and institutionalized activity should also be researched.

Second, it would be interesting to study the use of students becoming further involved in OSS usability work and in more detail than was possible in this thesis. There has been further UKKOSS projects in which student usability teams have been working in OSS development projects and the results and further analysis will be detailed in future papers.

Also, it would be interesting to fit usability cost-benefit considerations into other SW development contexts beyond commercial and OSS development contexts. There has been an increasing interest in researching usability in computer and console games and it would, therefore, be interesting to see how well the existing usability cost-benefit considerations fit into the game development context and how these considerations would have to be changed for the better.

Finally, one could study further how far the usability activities and processes can be adapted to the OSS development philosophy without compromising the core HCI philosophies, and how OSS development practices could be adapted to enable easier incorporation of usability activities without compromising the core OSS development philosophies. There has been a validation study with an OSS development project that is similar to UKKOSS 2 and has a similar approach to introducing usability. The usability team conducted testing and redesign, while the code team implemented the changes to some parts of the user interface and communicated the necessities of these changes to the core developers and community. The initial findings from this study indicate that the code team can more readily establish itself within the community and communicate usability problems, needs for change, and benefits of better usability to the core developers by backing up their arguments with already working code and, in this way, gain prestige among their peers.

References

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

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

Apache.org (2011) The Apache Software Foundation. URI: http://apache.org Cited: 2011/06/01.

Artman H (2002) Procurer usability requirements negotiations in contract development. In: Bertelsen, OW, Bødker S, & Kuutti K (eds.): Proc. 2nd Nordic Conference on Human-Computer Interaction. ACM Press, New York: 61-70.

Asaro P (2000) Transforming Society by Transforming Technology: the science and politics of participatory design. Accounting, Management and Information Technologies 10(4): 257-290.

Bach P & Carroll J (2009) FLOSS UX Design: An Analysis of User Experience Design in Firefox and OpenOffice.org. Proc. OSS 2009: 237-250.

Bach P, DeLine R & Carroll J (2009) Designers Wanted: Participation and the User Experience in Open Source Software Development. Proc. CHI 2009: 985-994.

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

Berquist M (2003) Open Source Software Development as Gift Culture: Work and Identity Formation in an Internet Community. In: Garsten & Wolff (eds.) New Technologies at Work, People, Screens and Social Virtuality, Oxford, Berg Publishers.

Bergquist M & Ljungberg J (2001) The power of gifts: organizing social relationships in Open Source communities. Information Systems Journal, 11 (4): 305-320.

Bevan N (2000) Cost Benefit Analysis. TRUMP report, September 2000.

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

Bias R & Mayhew D (2005) Cost-Justifying Usability, Second edn., An Update for the Internet

Age. Academic Press, Boston.

Boivie I, Gulliksen J & Göransson B (2005) The lonesome cowboy: A study of the usability designers role in systems development. Interacting with Computers 18: 601-634.

Buede D (2008) The Engineering Design of Systems: Models and Methods, Second Edition. John Wiley & Sons Inc., New York.

Bulkeley W (1992) Study finds hidden costs of computing. The Wall Street Journal, B4, November 2.

Burrill C & Ellsworth L (1980) Modern Project Management: Foundations for Quality and Productivity. Burrill-Ellsworth. New Jersey: pp 209-223.

Cajander Å, Gulliksen J & Boivie I (2006) Management Perspectives on Usability in Public Authority – A Case Study. Proc. of NordiCHI 2006. Oslo, Norway: 38-47.

Catarci T, Matarazzo G & Raiss G (2002) Driving usability into the public administration: the Italian experience. International Journal of Human-Computer Studies 57:121-138.

Cetin G, D. Verzulli & 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.

Champney R, Kokini C & Stanney K (2011) Making the Design Process More Usable: Aligning Design with User Performance. In: Marcus A (ed.) Design, User Experience, and Usability. Theory, Methods, Tools and Practice. Lecture Notes in Computer Science 6769, Springer, Berlin

Cockton G & Woolrych A (2002) Sale Must End: Should Discount Methods be Cleared off HCI's Shelves? Interactions sep/oct 2002: 13-18.

Conklin P (1991) Bringing Usability Effectively into Product Development. Human Computer Interface Design: Success Cases, Emerging Methods, and Real-World Context, Boulder, Co., July.

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

DIRKS (2003) A Strategic Approach to Managing Business Information: Appendix 10 – Recordkeeping Cost-Benefit Analysis. National Archives of Australia.

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

Due RT (1989) Determining Economic Feasibility: Four Cost/Benefit analysis methods. Journal of Information System Management 6(4): 14-19.

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

Feller J & Fitzgerald B (2000) A Framework Analysis Of The Open Source Development Paradigm. Proc. of 21st International Conference on Information Systems. December 10-13, 2000, Brisbane, Australia: 58-69.

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

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

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

Givnish TJ, Burkhardt EL, Happel RE & Weintraub JD (1984) Carnivory in the bromeliad brocchinia reducta, with a cost/benefit model for the general restriction of carnivorous plants to sunny, moist, nutrient-poor habitats. The American Naturalist 124(4): 479-497.

GNU.org (2010) Why Open Source misses the point of Free Software. URI: http://www.gnu.org/philosophy/open-source-misses-the-point.html Cited 2011/06/01.

Gould JD, Boies SJ & Ukelson J (1997) How to design usable systems. In: Landauer M, Landauer TK & Prabhu P (eds.) Handbook of Human-Computer Interaction, Elsevier, Amsterdam: 231-254.

Green C, Eiser A, Tollinger I, Castro L, Ratterman C, Vera A & Pyrzak G (2009) Leveraging Open-Source Software in the Design and Development Process. Proc. of the 27th international conference extended abstracts on Human factors in computing systems (CHI 2009): 3061-3074.

Greenbaum J & Kyng M (1991) Design at Work. Cooperative Design of Computer Systems. New Jersey, Lawrence Erlbaum Associates.

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

Gulliksen J (2000) Bringing in the Social Perspective: User Centred Design. CID-101, KTH, Stockholm, Sweden.

Gulliksen J, Göransson B, Boivie I, Blomkvist S, Persson J & Cajander Å (2003) Key principles for user-centred systems design. Behaviour & Information Technology on Designing IT for Healthy Work 22(6): 397-409.

Harrison M, Henneman R & Blatt L (1994). Design of a Human Factors Cost-Justification Tool. In: Bias R & Mayhew D (eds.) Cost-Justifying Usability. Academic Press.

Hedberg H, Iivari N. Rajanen M & Harjumaa L (2007). Assuring Quality and Usability in Open Source Software Development. Proc. of the First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS 2007), in conjunction with 29th International Conference on Software Engineering (ICSE 2007), 20 - 26 May 2007, Minneapolis, USA.

Himanen P (2001) Hakkerietiikka ja informaatioajan henki [The hacker ethic and the spirit of the information age]. Helsinki, Finland. WSOY.

Hughes M (2002) E-commerce Smash-Mouth. User Experence, UPA Jan 2002 (1)1: 16-18.

Iivari J & Iivari N (2011) Varieties of user-centredness: an analysis of four systems development methods. Information Systems Journal. 21(2): 125-153.

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

Iivari N, Hedberg H & Kirves T (2008) Usability in Company Open Source Software Context. Initial Findings from an Empirical Case Study. Proc. of the 4th International Conference on Open Source Systems: 359-365.

Iivari N & Molin-Juustila T (2009) 'Listening to the Voices of the Users' in Product Based Sofware Development. International Journal of Technology and Human Interaction. 5(3): 54-77.

ISO/IEC 9241-11 (1998) Ergonomic requirements for office work with visual display terminals (VDT)s – Part 11 Guidance on usability.

ISO/IEC 13407 (1999) Human-centred Design Processes for Interactive Systems.

ISO/IEC 9126 (2000) Software Product Quality – Quality Model.

Jokela T (2004) When Good Things Happen to Bad Products: Where are the Benefits of Usability in the Consumer Appliance Market? Interactions nov/dec 2004: 29-35.

Järvinen P (2001) On research methods. Opinpajan kirja.

Karat CM (1993) Cost benefit and business case analysis of usability engineering. Tutorial presented at the ACM SIGCHI Conference on Human Factors in Computing Systems. New Orleans, LA, April 28-May 2.

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

Klein HK & Myers MD (1999) A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly, Vol. 23(1): 67-94

Kujala S (2003) User involvement: a review of the benefits and challenges. Behaviour & Information Technology 22(1): 1-16.

Lakhani K & von Hippel E (2000) How Open Source Software Works: 'Free' User-to-User Assistance. Working paper 4117. MIT Sloan School of Management, Boston.

Landauer T (1995) The Trouble With Computers – Usefulness, Usability and Productivity. MIT Press, Cambridge.

Lazar J, Jones A & Shneiderman B (2006) Workplace user frustration with computers: an exploratory investigation of the causes and severity. Behavior and Information Technology 25(3), pp. 239-251.

Levy S (1984) Hackers: Heroes of the Computer Revolution. Anchor Press, New York.

Ljungberg J (2000) Open source movements as a model for organizing. Proc. of ECIS 2000.

March ST & Smith GF (1995) Design and Natural Science Research on Information Technology. Design Support Systems 15(4): 251-266.

Marghescu D & Rajanen M (2005) Assessing the Use of the SOM Technique in Data Mining. Proc. of IASTED International Conference on Databases and Applications (DBA 2005), Innsbruck, Austria.

Marghescu D & Rajanen M & Back B (2004) Evaluating Quality of Use of Visual Data-Mining Tools. Proc. of 11th European Conference on Information Technology Evaluation (ECITE 2004), Amsterdam, Netherland.

Mauro C (1994) Cost-Justifying Usability in a Contractor Company In: Bias R & Mayhew D (eds.) Cost-Justifying Usability. Academic Press, London: 123-142.

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

Mockus A, Fielding R & Herbsleb J (2000) A Case Study of Open Source Software Development. The Apache Server. Proc of ICSE 2000: 263-272.

Mozilla.com (2011) Mozilla Firefox Web Browser URI: http://mozilla.com Cited: 2011/06/01.

Nichols D, Thomson K & Yeates SA (2001) Usability and Open Source Software Development. Proc. of the Symposium on Computer Human Interaction (SIGCHI New Zealand): 49-54.

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

Nichols D & 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, & York P (2006) A Research Agenda for Studying Open Source I: A Multilevel Framework. Communication of the Association for Information Systems 18: 19-149.

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

Norman D (2002) Psychology of Everyday Things. Baskc Books.

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

OpenOffice.org (2011) The Free and Open Productivity Suite URI: http://openoffice.org Cited: 2011/06/01.

OpenSouce.org (2011) History of the OSI URI: http://opensource.org/history Cited: 2011/06/01.

Pemberton S (2004) Scratching Someone Else's Itch (Why Open Source Can't Do Usability). Interactions January + February: pp. 72

Rajala R, Rossi M, Tuunainen V & Korri S (2001) Software Business Models. A Framework for Analysing Software Industry. Tekes, Technology Review 108/2001, Paino-Center Oy, Helsinki.

Rajanen M (2002) Assessing the Business Benefits of Usability in the Product Development Project - Analysing the Existing Models. Proc. of 25th Information Systems Research Seminar In Scandinavia (IRIS25), Bautahöj, Denmark.

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

Rajanen M & Jokela T (2004) Analysis of Usability Cost-Benefit Models. Proc. of the 12th European Conference on Information Systems (ECIS2004), Turku, Finland.

Rajanen M & Marghescu D (2006) The Impact of Game Usability to Player Attitude. Proc. of 29th Information Systems Research Seminar In Scandinavia (IRIS29), Helsingoer, Denmark.

Rajanen M (2006) Different Approaches to Usability Cost-Benefit Analysis. Proc. of 13th European Conference on Information Technology Evaluation (ECITE 2006), Genoa, Italy.

Rajanen M (2007) Usability Cost-Benefit Models - Different approaches to Usability Cost Analysis. Proc. of the 9th International Conference on Enterprise Information Systems (ICEIS 2007). Funchal, Madeira - Portugal.

Rajanen M & Iivari N (2007) Usability Cost-Benefit Analysis: How Usability Became a Curse Word?. Proc. of the INTERACT 2007. Rio de Janeiro, Brasil.

Rajanen M & Iivari N (2010). Traditional Usability Costs and Benefits - Fitting them into Open Source Software Development. Proc. of 18th European Conference on Information Systems (ECIS 2010). Pretoria, South Africa.

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

Rolandsson B, Bergquist M & Ljungberg J (2009) Open Source Programmer's Strategies to Cope with Ideological Tensions. Proc. Organizational Learning, Knowledge and Capabilities (OLKC) 2009. Amsterdam, Netherlands.

Schaffer E (2004) Institutionalization of Usability: A Step-by-Step Guide. Addison Welsey. Redwood.

Schneider MF (1985) Why ergonomics can no longer be ignored. Office Administration and Automation 46(7): 26-29.

Schneiderman B (1998) Designing the User Interface: Strategies for Effective Human-Computer Interaction. 3rd ed. Reading: Addison-Wesley.

Singh V, Twidale M & Rathi D (2006) Open Source Technical Support: A Look at Peer Help-Giving. Proc. of the 39th Hawaii International Conference on Systems Sciences.

SourceForge.net (2011) Find, Create, and Publish Open Source software for free. URI: http://sourceforge.net Cited 2011/06/01.

Speare A (1971) A Cost-Benefit Model of Rural to Urban Migration in Taiwan. Population Studies 25(1): 117-130.

Spinuzzi C (2002) A Scandinavian Challenge, a US Response: Methodological Assumptions in Scandinavian and US Prototyping Approaches. In: Haramundanis K & Priestley M (eds.) Proc. 20th Annual International Conference on Computer Documentation. ACM, New York: 208-215.

Stallman R (1985) The GNU Manifesto. URI: http://www.gnu.org/gnu/manifesto.html Cited: 2011/06/01.

Standish Group (2008) Trends in Open Source. Standish Group.

Stewart K & Gosain S (2006) The Impact of Ideology on Effectiveness in Open Source Software Development Teams. MIS Quarterly 30(2): 291-314.

Szczepanska A. (2003) Understanding Open Source and Free Software from a Critical Perspective. Proc. of Open Source Software Movements and Communities workshop held September 19, Amsterdam, Netherlands.

Terry M, Kay M & Lafreniere B (2010) Perceptions and Practices of Usability in the Free/Open Source Software (FOSS) Community. Proc. of the 28th international conference on Human factors in computing systems, New York, NY, USA, ACM.

Tudor L (1998) Human Factors: Does Your Management Hear You? Interactions, Vol. 5(1): 16-24.

Twidale M & Nichols D (2005) Exploring Usability Discussions in Open Source Development. Proc. of the 38th Hawaii International Conference on System Sciences.

Wang SJ, Middleton B, Prosser LA, Bardon CG, Spurr CD, Carchidi PJ, Kittler AF, Goldszer RC, Fairchild DG, Sussman AJ, Kuperman GJ & Bates DW (2003) A cost-benefit analysis or electronic medical records in primary care. The American Journal of Medicine 114(5): 397-403.

Wardrip-Fruin N & Montfort N (2003) The New Media Reader. MIT Press.

Wixon D & Jones S (1991) Usability for fun and profit: A case study of the design of DEC RALLY version 2. Internal report, Digital Equipment Corporation.

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

Yin R (1994) Case study Research: Design and Methods. Second Edition. Sage Publishing. California.

Zeitlyn D (2003) Gift economies in the development of open source software: anthropological reflections. Research Policy, vol 32(7): 1287-1291.

Zetie C (1995) Practical User Interface Design – Making GUIs Work. McGraw-Hill Book Company, London.

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

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

Ågerfalk P & Fitzgerald B (2008) Opensourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. MIS Quarterly 32(2): 385-410.

List of original publications

This thesis is based on the following publications, which are referred to in the text by their Roman numerals.

I Rajanen M & Jokela T (2004) Analysis of Usability Cost-Benefit Models. Proceedings of the 12th European Conference on Information Systems (ECIS2004), Turku, Finland. ISBN: 951-564-192-6.

II Rajanen M (2006) Different Approaches to Usability Cost-Benefit Analysis. Proceedings of 13th European Conference on Information Technology Evaluation (ECITE 2006), Genoa, Italy. ISBN: 978-1-905305-32-2 (book) / 978-1-905305-34-6 (cd)

III Rajanen M & Iivari N (2007) Usability Cost-Benefit Analysis: How Usability Became a Curse Word?. Proceedings of the INTERACT 2007. Rio de Janeiro, Brasil. ISSN: 0302-9743, LNCS: 4663, ISBN: 978-3-540-74799-4, DOI: 10.1007/978-3-540-74800-7_47

IV Rajanen M, Iivari N & Anttila K (2011) Introducing Usability Activities into Open Source Software Development Projects – Searching for a Suitable Approach. Accepted to Journal of Information Technology Theory and Application 2011.

V Rajanen M & Iivari N (2010) Traditional Usability Costs and Benefits - Fitting them into Open Source Software Development. Proceedings of 18th European Conference on Information Systems (ECIS 2010).