Successful Usability Testing in Open Source Software Projects

University of Oulu
Dept. of Information Processing Sci.
Master’s Thesis
Matthew Ian Barth
16 May 2014
Abstract

Open source software has long been the domain of in-house production and personal use software programs. As the Internet becomes less and less of a specialist tool, this particular paradigm shifts. Now, many commercially available pieces of software are being created, as a whole or in part, with open source projects. As this happens, open source coders, who are used to creating software for themselves are facing the need to create a product that other people can use, some of whom are not technology aficionados. This thesis examines the process of implementing usability activities in open source software and looks at what does or does not work in the context of the open source community. This process begins with the creation of a usability case study, guided by Design Science Research method. Once that has been completed, a Comparative Analysis is conducted of that case study as compares to eight other cases from the same research series. As a result of this work, a methodology is proposed to outline the process of introducing usability testing into an open source project.

Keywords
Usability Testing, Open Source Software, Human-Computer Interaction
Foreword

I would really like to thank everyone, but, sadly, that is not feasible.

I do need to take the time to thank Adrian Zimmermann at Snowflake Productions for supplying UKKOSS with software to test, as well for offering any and all assistance along the way.

This thesis is based on the previous work of my advisor, Mikko Rajanen, and the various UKKOSS research teams, without whom the groundwork for this effort would still have to be laid. Therefore, I’d like to end this by thanking Mr. Rajanen for all his help in getting this thesis to where it is now.

Matthew I. Barth

Oulu, August 30, 2013
# Contents

Abstract................................................................................................................... 2  
Foreword.................................................................................................................. 3  
Contents................................................................................................................... 4  
1. Introduction........................................................................................................ 5  
2. Prior research.................................................................................................... 6  
   2.1 Usability Testing.......................................................................................... 6  
   2.2 Usability Philosophy................................................................................. 7  
   2.3 Open Source Software.............................................................................. 7  
   2.4 Open Source Culture............................................................................... 8  
   2.5 Usability in Open Source......................................................................... 9  
   2.6 UKKOSS.................................................................................................... 10  
3. Methodology..................................................................................................... 11  
   3.1 Design Science Research......................................................................... 11  
   3.2 Comparative Analysis............................................................................. 12  
4. Implementation.................................................................................................. 13  
   4.1 Case Study.................................................................................................. 13  
   4.2 Comparison of UKKOSS Projects............................................................ 18  
      4.2.1 UKKOSS 1.......................................................................................... 19  
      4.2.2 UKKOSS 2.......................................................................................... 19  
      4.2.3 UKKOSS 3.......................................................................................... 20  
      4.2.4 UKKOSS 4.......................................................................................... 22  
      4.2.5 UKKOSS 5.......................................................................................... 22  
      4.2.6 UKKOSS 6.......................................................................................... 23  
      4.2.7 UKKOSS 7.......................................................................................... 24  
      4.2.8 UKKOSS 8.......................................................................................... 24  
5. Findings............................................................................................................. 25  
   5.1 UKKOSS 10’s Analysis of todoyu............................................................... 25  
   5.2 UKKOSS 10 Compared to Its Predecessors.............................................. 28  
6. Discussion......................................................................................................... 32  
   6.1 Research Question.................................................................................... 32  
   6.2 Implications................................................................................................. 33  
7. Conclusions....................................................................................................... 37  
   7.1 Contribution............................................................................................... 37  
   7.2 Limitations................................................................................................. 37  
   7.3 Recommendations for Future Research................................................... 38  
References............................................................................................................. 39  
Appendix - Usability Test Report........................................................................... 44
1. **Introduction**

This work will explore how usability testing can be applied to open source software. That exploration starts with a look at the creation of a usability case study. It continues from there with a comparison of the results of that case study with those from eight additional cases from the same research series. This data will be used to draw conclusions as to what the best practices are for including usability testing in open source development.

This research could be very useful to the fields of software development, information systems research, and human-computer interaction. The need for this research arises because open source software practices are being used increasingly often to develop publicly used software and industry standard platforms. It is of the highest imperative that the culture surrounding open source development understands the value of usability, as that is the only way to ensure that new software is designed to best suit the tasks for which it is being created. Software is of no use to anyone if it cannot be easily utilized in the environment in which it will be deployed.

Before going into the study itself, this thesis will look at the history of the open source movement, usability testing, and the combination of the two in the UKKOSS research series. UKKOSS stands for “Ulkopuolisten Käytettävyys Konsultointi Vaikuttevuus Open Source Softwaressa,” which translates into English as, “External Usability Consultation Effectiveness in Open Source Software”. This history allows the reader to establish a context for the study and understand what has been done before. With this understanding, the reader can see that in the current state of the field, free software and open source software are not properly understood as separate entities, and usability is thought of as something of an afterthought, if at all.

By the end of this thesis, the reader should be able to answer the following question: “What actions lead to successes and failures when conducting usability testing on open source projects?”

Before answering the question the Design Science Research method, as outlined by Hevner (2004), is used to create a case study and establish an overall approach to improving usability. This establishes the author's knowledge base and gives a perspective from which the work can be taken as authoritative. After the completion of that investigation, a Comparative Analysis is conducted, further answering the research question (“What leads to success?”) using the first eight UKKOSS projects as a lens to examine what did and did not work. This second question fills a gap in what is known about the interactions between usability and open source software development.

Therefore, the aim of this thesis is to examine usability in an open source context and to provide concrete suggestions to usability professionals as to how they can operate in the open source community. In doing so, it will focus on past cases and, consequently, real world examples.
2. Prior Research

Usability testing in open source software did not simply spring from the void; there are many works over the years where its aspects are described. Though the combination of the two is a newer phenomenon, there is as rich history for both usability and open source software in information systems research. The UKKOSS project itself represents much of the academic work done to apply the principles and practices of usability testing to open source software projects.

2.1 Usability Testing

Usability Testing has already been given a set of standardized vocabulary. By doing so, the International Standards Organization has given the community working definitions of the terms to use. ISO 9241 Part 11 supplied the first definition, that of usability. Usability is “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (ISO, 1998). That definition offers researchers a more concrete foundation on which to test. In ISO 25010, another definition can be found, that of software usability. That phrase is defined as “[t]he degree to which the software product satisfies stated and implied needs when used under specified conditions (ISO, 2011). A decade later, ISO 9241 Part 210 added details for how one can go about designing software with a user centered approach, by looking at human-centered design activities namely: Understand and specify the context of use, Specify the user requirements, Produce design solutions, and Evaluate (ISO, 2009).

When put into practice, these design activities can then be used as criteria for testing. Usability experts outlined five quality components of usability: learnability, efficiency, memorability, errors, and satisfaction (Nielsen, 1993; Shneiderman, 1998). A major part of usability testing is usability inspection; Nielsen (1994) stated most succinctly, “Usability inspection is the generic name for a set of cost-effective ways of evaluating user interfaces to find usability problems.” For Nielsen, usability inspection was the first step to understanding the interface, as it could be done by a single evaluator or small team with minimal cost. That is only half the issue, as it has come to be accepted that without the input of users, the interface is highly unlikely to meet all the users’ needs (Holzinger, 2005). In the end, usability testing is the suite of tools and methods that can be used to find errors in user interfaces by conducting usability inspections and user tests to empirically locate as many problem areas as possible and then determine how they can be corrected.

From the definitions given, one can move on to what has been done in the information systems field. Though usability testing is cited as having been first recorded in the 1940’s, when Henry Dreyfuss user tested mock-up ocean liner staterooms (Arent, Arnowitz & Berger, 2010), it didn’t find its way into the software literature until the 1979, with John Bennett’s The Commercial Impact of Usability in Interactive Systems (Bennett, 1979). Usability was further legitimized in 1982, when the Special Interest Group for Computer Human Interaction was formed, as well as the following year, when the first ACM SigChi Conference was held in Geneva (Shneiderman, et al., 2002). At that conference, the first paper on conducting usability testing was presented by Bewerly (1983), outlining the process of creating an interface for the Xerox “Star”
workstation. Whiteside, Bennett & Holtzblatt (1988) made the case for moving usability from a topic of academic interest to a full fledged profession. Due to the new demand from usability professionals, the 1990's brought the introduction of new methodologies, which are still used today, in the form of Cognitive Walkthrough (Lewis, Polson, Wharton, & Rieman, 1990) and Heuristic Evaluation (Nielsen & Molich, 1990). Usability, at this point, has been shown to be a worthwhile endeavor, as set out in Bias & Mayhew's (1994) book, Cost Justifying Usability.

2.2 Usability Philosophy

Iivari & Iivari (2011) have identified four ways researchers have looked at user-centeredness since the late 1980's. In their work, it was seen that user-centeredness could be examined in the light of user focus, work-centeredness, user involvement or participation, and system personalization. Each of these four viewpoints approach research and its results in a different manner. This shows the research community that it is important to not just include users-centeredness, but to think about how it is included and what that will do for the final product or system being tested.

The differences in research styles can be seen by looking at different periods and their respective philosophies. Earlier researchers knew that something had to be done to improve usability, but the thinking at the time was far more clinical and detached, as seen in Gould and Lewis (1985): “Early focus on users and tasks” was a part of the formula, but it was included by, “directly studying their [users’] cognitive, behavioral, anthropometric, and attitudinal characteristics, and in part by studying the nature of the work expected to be accomplished.”

In current literature, usability activities are best represented in Human-Computer Interaction. First and foremost, usability specialists need to remain user-centered, as Rajanen & Iivari pointed out the following in their MindTrek paper, which stated:

> Usability Specialists are user representatives. They act in informative, consultative, participative and designer roles, sticking with the user focus, knowing the user, speaking for and fighting for the user in the development [process]. [All while] having decision-making power regarding important design solutions.

(Rajanen & Iivari, 2013, p. 2)

It is this focus that keeps modern specialists working on the human side of Human-Computer Interaction.

2.3 Open Source Software

The term “Open Source Software” is widely known, but what it implies is less common knowledge. It might be helpful to begin by looking at how Free/Libre and open source differ. The Free/Libre movement looks at the creation and use of software as a human right, like free speech. While Open Source shares a similar theory, it manifests as a pragmatic development methodology (Crowston et al., 2005). Open Source Software focuses on the free redistribution of source code while maintaining the integrity of each author's work, keeping the process free from discrimination (opensource.org/osd). Free Software does this with a different focus; they are looking to “make a political and ethical choice asserting the right to learn, and share what we learn with others” (fsf.org). Both can be encountered in the culture, and it is important to know under which
umbrella a given project falls, so as not to make false assumptions regarding the goals of any given project.

Like any subculture, the open source software community has a rich history. A key launching point for the open source movement was Richard Stallman’s GNU Project, launched in 1983 (Stallman, 1983). Frustrated with his inability to fix a software-driven piece of equipment he owned, Stallman went on to author the GNU Manifesto over the next two years (Stallman, 1985). Over time, the movement gathered strength, and culminated in Linus Torvalds’ famous project, the Linux Kernel, which enabled an entire system to be developed using nothing but free software (Torvalds, 1997). The development community for Debian took this project philosophy to heart, crafting a distribution of the Linux operating system that was built solely on free software, and could be obtained by anyone (Debian Project, 2004).

The term Open Source wasn’t coined until the late 1990’s. It came about as a term to describe the release of Netscape’s code to the Navigator web browser, as they had opened the source code to public use in an attempt to keep the company afloat (Bonaccorsi & Rossi, 2006). The term gained ground when it was picked up by Linus Torvalds and later by Tim O’Reilly’s Freeware Summit, now retitled the Open Source Summit (O’Reilly, 1999). This has left Open Source Software where it is today, where the focus is more on open access to source code than an appeal for software freedom (fsf.org).

2.4 Open Source Culture

Entry into the open source community requires an understanding of the community itself:

[S]ocio-technical structures are important because anyone interested in contributing to the production of a [Free/Libre or Open Source Software] project must learn to negotiate the structures in order to participate. Social structures involve the skills and procedures necessary for contribution.

(Bach & Carroll, 2010, p. 903)

Since Open Source Software developers have often, historically, been developer/users, they create software that runs the way they want to use it (Nichols & Twidale, 2003). Software developed in an open source community can be allowed to grow based on the needs of the developers themselves, since they are the ones that will be using the product (Crowston & Scozzi, 2002). By working as a collective, open source developers can let the community solve old problems while they themselves move on to solve new ones that they find interesting (Raymond, 1999).

The open source community has long been seen as a meritocracy. Many people have worked to discover how this plays out in actual projects. For example, it was shown by O’Mahony & Ferraro (2007) that maintaining frequently installed software packages increases the likelihood of a developer moving up in the ranks of a project, but simply pushing many changes had an inverse effect on advancement. This shows that it is the quality of contributions that are important to the community’s perceptions of its members. The same research team showed that there was no evidence that on-line communications created an awareness of the contributor’s management potential, but members who met with the leadership face-to-face were 56 percent more likely to be given positions of responsibility (O’Mahony & Ferraro, 2007). This shows a desire
within the community for fostering trustworthy relationships and identifying people who are willing to actively work with others.

In defining personal merit, seven basic values have been identified in the Open Source software movement: sharing, helping, technical knowledge, learning, voluntary contribution, and reputation (Steward & Gosain, 2006). It is values, such as these, coupled with one or more strong coordinators, that hold open source communities together (Ljungberg, 2000).

Acceptance of non-technical contribution has been unofficially seen as a difficult barrier to cross, although not a lot has been said about it in the literature. Occasionally, a mention of it can be found, such as an anecdotal mention in the a case study of the GNOME Project, where a writer and a graphic designer were able to secure positions in the hierarchy (German, 2003). Though, when talking about the same project, another research group noted that: “Dedicated usability mailing lists and IRC channels do exist but they host few useful discussions and, along with the complex bug database, can be intimidating to non-technical contributors” (Benson, Muller-Prove & Mzourek, 2004).

2.5 Usability in Open Source

There seems to be a divide between the desire to create user-friendly software and the willingness to include usability as a priority in development:

> Overall we found that OSS developers are interested in usability, but in practice most of the efforts are based on commonsense. They appreciate external usability evaluations performed by professionals, as long as these professionals are not interfering in decision-making about changes and priorities.

(Andreasen, et al., 2006, p. 303)

Bødker et al. (2007) worked to find a solution to this divide, working to increase user awareness and usability activities in open source development. The precarious relationship between open source development and usability was also studied in relation to social context (Zhao, 2005). It has been shown that this gap is primarily found between the design and implementation of the project; everyone has their own ideas, and design choices get changed as development progresses (Moghaddam, et al., 2011).

Other researchers are also investigating phenomena related to usability in free or open software. Schwartz and Gunn (2009), for example, made efforts to bring attention to usability's status as “largely an afterthought” in free/open source software's user experience design, of which usability is a category. Another recent advance in the usability of open source software field came from Terry, Kay and Lafreniere (2010), who claim their results “argue for the need to research new [human-computer interaction] methods that operate in the culture and value system of the [free/open source software] community.”
2.6 UKKOSS

Open source software has been known to produce software that, at times, can be cryptic to many of its users. To gain a better understanding of this phenomenon, UKKOSS, a research initiative in the University of Oulu [Finland]'s Information Processing Science department, focused on exploring the use of usability testing in open source software.

At the time of this publication, ten UKKOSS projects have been conducted and eight of them have had information published about them. UKKOSS 1 through 4 were included in *Introducing Usability into Open Source Software Development Projects – Searching for a Suitable Approach* (Rajanen et al., 2011). UKKOSS 5 through 8 were later published in *Introducing Usability into Open Source Software Development Projects – a Participative Approach* (Rajanen et al., 2012).

Still, UKKOSS is not doing its work in a vacuum. Much of the UKKOSS research has been conducted based on the principles and methods of Design Science Research (Hevner et al., 2004; Peffers et al., 2007). Additionally, the usability activities the research teams use have been tried and tested by others before them (Nielsen, 1994; Federoff, 2002; Desurvire, 2004; Holzinger, 2005; Pinelle, 2008).
3. Methodology

This work is created a case study and conducted a comparative analysis, which were used to gain a greater understanding of how one can conduct usability testing in open source software projects. The first stage utilized design science research, and involved collecting data during the process of testing and improving the user interface of a piece of web-based, task and project management software, called todoyu, which is being developed under the guidance of the Swiss firm Snowflake Productions. In a collaborative effort between UKKOSS and Snowflake Productions, progress was determined based on the answers to the question, “How can one conduct usability research on an Open Source project without compromising the process or quality of the product?” The second stage employed a comparative analysis, contrasting UKKOSS 10 with its previous incarnations, at this stage asking the real question, “What actions lead to successes and failures when conducting usability testing on open source projects?” Though it shouldn't be necessary, it bears a reminder that usability testing should be done without compromising the process or quality of the product being developed.

3.1 Design Science Research

The core of the research for this work was conducted during the implementation of UKKOSS 10. At that time, a group of five students in the University of Oulu's Information Processing Science department conducted usability testing on a piece of Resource Management software being developed by Snowflake Productions.

Since Snowflake Productions is based out of Switzerland and The University of Oulu is in Finland, the research team joined talents with Snowflake's staff — as off-site consultants. As such, they worked together to identify and correct usability issues in the previous version of the todoyu platform, helping shape the next iteration of Snowflake's software. While collecting data, UKKOSS 10 focused on a problem that needed to be solved for them, which was: How can one conduct usability research on an Open Source project without compromising the process or quality of the product? (Please note that this question was internal to the UKKOSS study and should not be confused with the research aims of this paper.)

Evaluation of the todoyu software platform was conducted with the aid of Cognitive Walkthrough (Wharton, Rieman, Lewis, & Polson, 1994), Heuristic Evaluation (Nielsen & Molich, 1990) and User Testing (Nielsen, 2003). These methods were chosen based on having the longest standing in the usability testing community and adequate evidence to support their ability to provide value to the client. The Cognitive Walkthrough method was used to identify how the software package would be used by its various user groups and what types of tasks would need to be tested as the project progressed. Heuristic Evaluation was then conducted as a means of identifying what the testing team saw as the key usability issues with the todoyu software platform. Because usability inspection has been shown to miss some interface issues when used on its own, the team also conducted a series of user tests with the tasks have been created using the Cognitive Walkthrough method.

To maintain academic rigor, the project team utilized the Design Science Research process pioneered by Hevner:
Design as an Artifact - Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.

Problem Relevance - The objective of design-science research is to develop technology-based solutions to important and relevant business problems.

Design Evaluation - The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.

Research Contributions - Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.

Research Rigor - Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.

Design as a Search Process - The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.

Communication of Research - Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

(Hevner et al., 2004, p. 83)

In addition to the artifact, an approach to improving usability in open source software projects, the study provided assistance to Snowflake Productions in the usability of their current open source software project. Rigor was assured through iterations of testing and verification with another usability expert, Mikko Rajanen. The search process consisted of performing user tests until there was enough data to support the changes which needed to be made in the next release of todoyu. In the end, this body of work, combined with reports to the University of Oulu and Snowflake Productions, was designed to act as the agent of communication for the Design Science Research portion of the study.

3.2 Comparative Analysis

After the completion of data collection in UKKOSS 10, a comparative analysis was conducted using the documentation of UKKOSS 1 through 8. The guidelines for this segment of the study were taken from Harvard University's Kerry Walk (harvard.edu, 1998). The purpose of conducting a comparative analysis, in this study, was to identify the points where the various usability test processes which met with success and/or failure, and how those points relate to the fact that the projects were open source in nature.

All of the data relating to the past UKKOSS projects came from the works of Mikko Rajanen. His contributions to the topic offered the grounds upon which UKKOSS 10 was based, and he oversaw the work done in the Case Study. Therefore, the prior studies within the same series offer the closest basis for comparison.
4. Implementation

The studies contained in this thesis were conducted by the UKKOSS project group at the University of Oulu, Finland, during the Spring of 2013. Project members were taken from the Usability Testing course, and had already demonstrated knowledge of usability testing methods and theories, as this is the practical culminating course in a series. Project members each conducted an average of 200 hours work on the project, which was their first major work as newly trained usability testing experts.

4.1 Case Study

The first task of the UKKOSS 10 team was to find a suitable open source project with which to conduct usability testing. A pool of ongoing projects was compiled by the group and then narrowed down based on the following criteria: project size, project structure, usability need, and openness to outside teams. A project with a small to medium sized core team would, presumably, insure that the project had enough people to keep the activity level high on development, but not enough current staff to cover all of its testing needs. The structure of the project needed to have a community of developers, so that it wasn't just open source in name, and the researchers would be welcomed in. Additionally, the project developers needed to have a need for external help; several projects looked good on paper until it was realized that there were already hundreds of contributors to the user interface development alone. Based on all this, Snowflake Productions’ todoyu project management platform was chosen based on being the best fit. It has a small team of staff developers to keep the project flowing. There were signs of a modest group of user/developers in its community site and forums. To top it off, they were actively looking for a tester or testing team for the next release, making them a stand-out candidate.

After selecting a suitable project, the team had to collect as much knowledge as possible about todoyu and its users. User groups were identified from the forums, and the software itself was inspected, first informally, then in detail with Cognitive Walkthrough and Heuristic Evaluation (Nielsen, 1994; Wharton et al., 1994). Cognitive Walkthrough is conducted by a group of researchers who look at a system with the following criteria in mind:

1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. User control and freedom
Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

(Nielsen, 2005, p. 1)

Heuristic Evaluations, in turn, shift the focus from what the researchers are looking at to the method by which they examine the system.

1) The user sets a goal to be accomplished with the system (for example, “check spelling of this document”).

2) The user searches the interface for currently available actions (menu items, buttons, command-line inputs, etc.).

3) The user selects the action that seems likely to make progress toward the goal.
4) The user performs the selected action and evaluates the system’s feedback for evidence that progress is being made toward the current goal.

(Rieman et al., 1995, p. 387)

In this study, the Cognitive Walkthrough was conducted outside its proscribed usage. By this, it is meant that the process was followed, but results were not as heavily documented. The walkthrough was conducted simply to inform the researchers and improve their personal understandings of the system being tested. It was used as a tool to formulate how various users might desire to interact with the system, since the set-up for the walkthrough involved (1) a general description of who the users will be and what relevant knowledge they possess, (2) a specific description of one or more representative tasks to be performed with the system, and (3) a list of the correct actions required to complete each of these tasks with the interface being evaluated” (Rieman, et. al., 1995). After this inspection, there was a series of tasks that were identified:

- Add companies to the database
- Add staff members to the database
- Give rights to use functions and areas [i.e. calendar] in the system
- Create a new project
- Create a new task
- Use time tracking
- Set a weekly meeting
- Change the active user’s password
- Add a file to a project
- Add a comment to a task
- Search for a contact from the database

This list was then used as the script for the Heuristic Evaluation to follow.

The team used Nielsen’s ten usability heuristics (Nielsen, 1994) to evaluate different tasks in todoyu. Each team member along with 5 other participants went through a set of tasks multiple times to record usability issues and the corresponding heuristic violations that occurred. The issues were then rated in terms of severity. The severity scale used is as below:

- 0 - Don’t agree that this is a usability problem
- 1 - Cosmetic problem
- 2 - Minor usability problem
- 3 - Major usability problem; important to fix
- 4 - Usability catastrophe; imperative to fix

(Nielsen, 1994, p. 36)

After the Heuristic evaluation was completed, a pilot test was run to see if someone new to the software platform, todoyu, could complete the tasks unaided. This pilot caught a number of errors in the wording of tasks and helped recognize the order in which the tasks should be completed. As such, the tasks were re-written to accommodate the time allotted for each test session (one hour), as well as for clarity. The team's goal was to test the system, not to trip up its testers with tricky wording.

The user testing phase was conducted next, in the form of exploratory testing. Several different data collection methods were used for the testing, including observation, interviews, video analysis and thinking aloud. Test participants evaluated the system by going through a list of predefined tasks on a laboratory computer during a testing session. For each test user, the facilitator first briefed the test participant regarding the system and asked a series of pre-interview questions before the testing session started. This gave the tester context and also provided the research team with basic information
about the testers. Post interview question were asked after the testing session. This gave the researchers clearer insights into the testers' experiences and observations during the test session.

There were five testers participating in the actual testing phase. They had backgrounds in ICT, education, and business in their working lives. The participants all had prior experience in project work, so they were familiar with project tasks and also had some understanding of project management. The participants had good computing skills. Testers were given the anonymous monikers (P1, P2, P3, P4 and P5) to protect their privacy. P2, P3 and P5 had prior work experience in ICT sector. P1 and P4 were ICT students. The goal was to recruit testers that possessed solid computer skills, but with at least one slightly below and one advanced user. This gave the researchers the ability to record how less than ideal conditions would play out. The target market for the actual TODOYU platform consists of project managers for small-to-medium-sized projects and their employees. In this context, the role of a project manager is to create a project, manage project team members, maintain a project schedule and its related tasks, facilitate communications, and create needed reports and documentation.

The test participants were all between twenty and thirty years of age. Testers were recruited from the University of Oulu, through recruitment by the research team. The agreements with test participants were made orally. Testers were offered small rewards for their valuable contributions for the usability testing, like coffee and chocolate, after they completed the test session. None of the participants had ever heard of the TODOYU project management platform prior to testing it with the research team.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Age</th>
<th>Gender</th>
<th>Occupation</th>
<th>Relevant experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot</td>
<td>30</td>
<td>F</td>
<td>Student</td>
<td>Project management and group work</td>
</tr>
<tr>
<td>P1</td>
<td>26</td>
<td>M</td>
<td>Student</td>
<td>Project management work and has used Microsoft Project</td>
</tr>
<tr>
<td>P2</td>
<td>23</td>
<td>M</td>
<td>Web Designer</td>
<td>Has used Trello.com</td>
</tr>
<tr>
<td>P3</td>
<td>26</td>
<td>M</td>
<td>Game Designer</td>
<td>Project management work and uses non-software management tools</td>
</tr>
<tr>
<td>P4</td>
<td>25</td>
<td>M</td>
<td>Student</td>
<td>Has used targetprocess</td>
</tr>
<tr>
<td>P5</td>
<td>27</td>
<td>M</td>
<td>Entrepreneur</td>
<td>Project management work</td>
</tr>
</tbody>
</table>

The test scenarios were provided as role-playing scenario task lists for each of TODOYU’s three predefined roles: system administrator, project manager, and project team member. The role play scenarios motivated the test participants to do the testing and to immerse themselves into the testing situation. Each testing session was between 40 to 60 minutes in length, depending on the testing task and the test participant. The tasks were provided, step-by-step, in the form of printed slips. The list below shows the three different scenarios and the list of tasks contained in each.
**Scenario #1  Admin with system administrative activities**

Task A: The ICT Company, Rapid Testing, uses *todoyu* as their project and task management system for their daily project activities, and you are the administrator of Rapid Testing's *todoyu* system. To start your job, add Rapid Testing and their partner, Adelphi, as companies in the system.

Task B: You need to staff your projects, so add Mr. John Smith as a customer, working for Adelphi, and add Ms. Sirpa Doe as a project member, at Rapid Testing. Give them access to the *todoyu* system.

Task C: These people want to use the calendar. Give Mr. Smith the right to use the calendar's functionalities, and give for Ms. Doe the right to use the calendar area.

**Scenario #2  Project manager with project management activities**

Task D: You are now working as the project manager for one of Rapid Testing's OSS projects, and currently, you have several different projects and tasks to manage. Create a project called “OSS,” with the simple description of "First project." This project is in progress, since it started on 14.01.2012 and doesn't end until 15.05.2013. Adelphi is the customer for this project, with John Smith acting as the contact, and Sirpa Doe is going to lead the project from Rapid Testing's end.

Task E: You wanted to give a specific task to one of the project members. Add a “usability testing” task to the project. This task starts on 14.01.2012 and has to be to the customer by 15.05.2013, assign the testing task to Sirpa. It is expected that the task will take Sirpa 100 work hours to complete.

Task F: You want to track the time spent on specific tasks in order to send bills to customers. Start time tracking.

Task G: To stay on track, you want to schedule a “weekly meeting” event in the calendar every Wednesday, starting date on 13.03.2013 and an ending on 29.05.2013, from 15:00 -16:00 hours.

**Scenario #3 - Project member with regular project task activities**

Task H: You want to change your password in *todoyu*. Change your password from “PassworD123” to “ABc123”.

Task I: You have completed your OSS testing report, and you want to add the test report file to the usability testing task interface. Add a test report file to OSS project.

Task J: You want to inform your co-workers about the usability testing task report.

Write comment in the usability-testing task that says, “I have done the introduction and objective sections of the usability testing report.” Link the report to this note.

Task K: You have many different tasks from different companies, and you wanted to search Inferno Studios to find out its contact person. Use the global search to find a company starting with “Infer,” and use the results to view its contact person.
After the original script from the Usability Inspections was edited for flow, clarity, and bias reduction, the tasks were used in the testing sessions.

The testing laboratory used in this study had a laptop for usability testing, equipped with two web cameras to record the testers’ activities. Behind a barrier, there was an observation computer in the testing lab with which to record the screen of the testing laptop and the face of the tester. The objective was to observe the testing sessions without influencing the tester. The testing laptop was also utilizing Camtasia, a screen capture program, to record videos of the testing process.

4.2 Comparison of UKKOSS Projects

One researcher per session acted as a moderator, and any other researchers present observed the usability testing from behind the screen. The cameras attached to the testing laptop recorded the test sessions and the test videos were saved on an external hard drive for review. During the usability test sessions at least one of the observers acted as a data logger, collecting notes of things they saw as an observer that the camera might not have caught. The role of the test moderator was to interview and instruct the test participants, control the pace of the test session, and manage the testing sessions in general. The test observer was seated quietly on the other side of the testing lab to observe the activity of the test participant taking notes.

The testers were welcomed appropriately to the test sessions by the research team. The moderator was responsible for instructing and supporting the testers during test sessions. The moderator also provided the task lists for the test participants, one question at a time. Moreover, the moderator encouraged the testers during the test sessions. During test sessions there was no interaction between the other parts of the research team and the testers. The research team used interview and observation data gathering techniques to support test results. They observed the test participants’ activities as the testers interacted with the system during the test sessions. Semi-structured interviews were conducted before the test session to understand the background information of the test participants, and another longer post-test session interview was conducted after the test session to understand the testers’ impressions and experience after using the todoyu platform.

Using the guidelines for comparing bodies of work outlined by Kerry Walk (harvard.edu, 1998), a comparative analysis was conducted of the case study of Snowflake Productions’ todoyu by UKKOSS 10 and the prior UKKOSS projects. The grounds for this comparison are quite strong, as the supervisor of all of the UKKOSS projects was the same, Mikko Rajanen. Therefore, the bodies of work form a reasonable approximation of a longitudinal study. Since information has already been published on the first eight UKKOSS projects, they were used, to the exclusion of UKKOSS 9, which has not -to date- been documented outside of its research team.

This portion of the study ultimately tries to answer the question, “What actions lead to successes and failures when conducting usability testing on open source projects?” The simplified frame of reference, that there is some factor that makes testing succeed in some cases and not in others, was adopted for the comparative analysis.

Due to the nature of this study, its author only possesses first hand knowledge of the research on UKKOSS 10, and thus has far less information available to him on the earlier projects, UKKOSS 1-8. Because of this restriction, the information known about those projects was used as a lens through which the author could identify additional characteristics of UKKOSS 10 itself.
The data for the first eight UKKOSS projects comes from the studies published by Mikko Rajanen, Netta Iivari, et al. (2011, 2012). It has been reformatted here so that the reader can see the assumptions made, the projects chosen, the methodological choices, and the findings of each in turn, even though they are not themselves original findings of this thesis.

4.2.1 UKKOSS 1

It was assumed that usability experts could introduce their activities into open source software development by acting as consultants. External usability specialists could then act in a role in which they provide feedback to the development team, allowing them to come up with the actual solution. This was based on the belief that unsolicited usability feedback had a good chance of being taken to heart, as it may be seen as a wake-up call. Much of open source software relies on patches and features being added by the community at large, having usability added in the same way was not seen as a great departure from the norm.

In UKKOSS 1, the software being tested was for a media-center. The specific choice of project was made because it was active and large enough to remain so, while being small enough to be approachable. Additionally, the project cited its target demographic as being ordinary users, not experts. The project had about thirty developers and 16,000 users.

The research team acted as external usability consultants. They conducted usability inspections and user tests, afterward reporting the results to the project team, without contacting members prior to submitting the report. The usability activities were planned and conducted under supervision of a professional Usability/Human-Computer Interaction researcher, which ensured the quality of the usability results obtained. Different kinds of usability evaluations were chosen, based on widely used and successful methods documented in usability literature: Heuristic Evaluation, using Shneiderman’s heuristics, and Cognitive Walkthrough (Shneiderman, 1987, Wharton et al., 1992). Six users were brought in to test the software. After the usability findings were reviewed by the professional researcher, they were sent by email to the project developers.

No immediate response was received from the developers. Eventually, the usability report and summary of findings were posted to the project’s discussion forum in an unaltered state for the community to respond. In that space, the developers noted that they were discussing the results internally. No further answer has appeared and there is no sign of any impact on development, nor any sign that usability has gained any importance to the development team.

4.2.2 UKKOSS 2

After the results of UKKOSS 1, it was assumed that usability activities should, instead, be introduced by a participative approach. Usability specialists should try to become recognized and appreciated by the core developers and by the whole community, after which they hoped to be allowed to take an active part in the design process with the developers. It was seen as important for the researchers to prove they understood the software being developed and how it was to be utilized by its user base. The research team planned to perform usability activities that were similar to those in UKKOSS 1, but with changes as to how they would inject themselves into the project.
The project chosen for UKKOSS 2 was a game. It was selected because it was an active open source project of suitable size, with a target demographic of ordinary people, so as to be similar in scope to UKKOSS 1. The project selected had about fifteen developers and about 1000 users.

The developers of the game were contacted at the very beginning of the study, before any usability activities were conducted. The goal of that communication was to become members of the development community and directly work on the developers' attitudes toward the research team and their work. The developers therefore knew that the research team was going to be doing usability inspections and testing. They offered the developers any usability assistance they might need. The key assumption was that the developers would consider the input from the research team if the team was already a participant with an identity within the community. When the research team conducted their Heuristic Evaluation of the game, they combined several known game heuristics (Desurvire et al. 2004, Federoff 2002). A report of usability findings and consequent recommendations, similar to the one created by UKKOSS 1, was written. The number of usability findings and recommendations, as well as their severity, was similar to those reported by UKKOSS 1, although the suggestions for change in UKKOSS 2's documentation included redesigned user interface mock-ups. The research team sent the developers an additional document in which they introduced themselves and described their background and research goals. After the evaluation had been received, the developers were asked what additional work the research team could do for the project. The developers reported that they did not know what to do next, but they knew that they wanted their level editor to work in a way that anybody could quickly and easily pick it up. Based on this goal, the research team conducted another Cognitive Walkthrough, this time of the level editor, and made sketches for an improved user interface.

Utilizing the project’s IRC, an interview was conducted, which determined that the usability activities of the research team had improved usability awareness. The feedback from the research team appeared both welcomed and taken seriously. Mixed opinions were expressed about communication level. The lead developers would have liked increased communication and more activity on the project’s IRC channel, while other developers felt that the distance kept the researchers objective. Developers on this project readily welcomed the possibility of further usability specialists becoming part of their development team. Some negative feedback came as a result of the game’s development cycle; usability findings were sometimes obsolete by the time the reports were submitted to developers, since they were actively fixing bugs. Despite their attempt to become full members of the community, the research team members were ultimately treated as external consultants. Despite findings being added to the game's development wiki, the team was not allowed to include usability activities into the overall development plans themselves. The developers only considered the issues they, themselves, saw as problems. Problems not agreed upon by core developers got low priority.

4.2.3 UKKOSS 3

The approach taken in UKKOSS 3 was again participative, but this time in a large and complicated open source project with an expert user base. The goal of the research team was to become included in the community in order to see if the complexity of the hierarchical structure and/or expert user population would affect communications with developers and developers' attitudes toward the work the research team was doing. Usability specialists made the assumption that they should try to become recognized and appreciated by both core developers and the development community as a whole. The
hierarchical structure of large open source projects was expected to increase the difficulty in participating in the project in a way that recognition could be achieved, and would likely have an impact on the selection of suitable approaches when introducing usability activity into the project. To this end, UKKOSS 3 planned to perform a similar set of usability activities to previous projects, while trying to put more emphasis on getting the research team included as a part of the development community at the very beginning. A member of UKKOSS 1 and a project manager from the UKKOSS 2 joined the research team of UKKOSS 3, to improve the overall expertise of the research team.

The project chosen for UKKOSS 3 was free open-source 3D content creation software. It was a large project which included a complex hierarchical structure and many active sub-communities, each with their own leaders, websites, forums, etc. The software itself was also quite complex and had a target demographic of end users and organizations with high-level domain expertise. The project had about forty developers and about 800,000 users.

The research team started by familiarizing themselves with development history, the hierarchical structure of the project, core developers, and potential usability champions by following the projects forums, to-do lists, roadmaps, IRC channels, and development wiki. Congruent with traditional software development, the team submitted usability evaluations to act as a wake-up call within the open source project (Schaffer, 2004). The researchers encountered an inconsistency in the development process, namely that the developers were adamant about the user interface, while simultaneously having a stated goal of attracting talented experts to this system from commercial alternatives. The research team saw this as an opening that could be used to promote their cause within the community. Because the software also had reputation of being very difficult to learn and use, yet was seen as efficient after the steep learning curve was overcome, changes needed to be made to the user interface in order to make it more intuitive. The usability research team was able to show that the software was not yet intuitive enough, by pointing out the user requests for features that were already implemented, but remained unrecognized by users.

When conducting the study, UKKOSS 3 split into two research teams. One team handled discussions with the development community in order to gain an identity with the core developers and community and use that identity to promote the usability activities and results of the other team. The second team conducted the usability activities, which included user tests with four potential expert users. Although usability inspection methods were also used, the results were not practical, as the project was too complex and not suited to standard heuristics.

The user tests, on the other hand, resulted in a great deal of usability information, in the form of both issues and points of superiority to commercial alternatives. Test users expressed interest in transferring to the use of this software, but only if the user interface improved and industry conventions, such as shortcuts, were added to aid the changeover process. Comments like these were used as key arguments in the results, since these usability benefits matched with the goals of the core developers and the community. To maintain credibility within the community, results from the usability tests and other usability inspections, including a report of usability findings, recommendations, and mock-ups, was published open source on a usability team website. That website was then used to promote its findings in community forums, IRC channels, and community news sites. The community news website was down during the time what the research team wanted to publish their findings, delaying the results for critical months and burying the findings. Though usability improvements were evident in the project,
UUKKOS 3 failed to capture the attention of the development community that it had wanted. After the researchers stopped posting, no further usability discussion continued.

4.2.4 UKKOS 4

This study marked a return to a consultative approach to usability involvement in an open source project. The researchers planned to repeat the usability tasks and phasing of UKKOS 1 as closely as possible. This was done to see if the original results could be validated.

The project chosen for UKKOS 4 was another piece of media center software under active development with a target demographic of ordinary people. The project was large enough to be under constant development, but small enough so that the core developers were both identifiable and approachable. It had about twenty active developers and more than 700,000 users.

Usability inspections were taken from project report of UKKOS 1 so that they remained constant. In the same fashion, user testing was planned using the results of the usability inspections, congruent with UKKOS 1. Four user tests were conducted, using tasks as similar as possible to those in the prior study while still being applicable to the new software. All usability findings were compiled as a report, which described in detail the types of usability tests carried out, identified software versions tested, and gave improvement suggestions. As in UKKOS 1, this report was sent by email, as the first contact with developers, as would be done with software patch submission in the normal course of open source software.

The result of the study was that developers sent an acknowledgment of receipt but no further communication. The report was never discussed or mentioned in the project discussion forums, chat, or mailing, and no sign of any impact has been found relating to usability issues and redesign solutions reported. Given that result, the findings of UKKOS 1 appear to be valid.

4.2.5 UKKOS 5

Due to the analysis of UKKOS 1 through 4, evidence seemed to support the belief that usability activities should always be introduced by a participative approach. That approach was therefore taken by UKKOS 5. In UKKOS 5, there was an attempt made to select a project with no corporate ties, as it was assumed that if a corporation were behind the development, usability testing would be covered in-house. In the project eventually chosen, one of the usability specialists was already recognized and appreciated by the core developers and was therefore in a position to continue taking an active part in the design process. This created a new scenario to study, but it was still important for the rest of the researchers to prove they also understood the software being developed and how it was to be utilized.

The software to be tested by UKKOS 5 was a game in the single player rogue-like genre. It has been reworked by a rotating development team since 1995. The choice of project was made because it pointedly mentioned its user interface as being a key point of the software, and it claimed to be user friendly to new users. As a game, the project has a target demographic of gamers, not necessarily experts. The project had about twenty developers.
Many of the methods needed in previous UKKOSS projects were not as necessary in UKKOSS 5, as a member of the research team was already in the project and usability was already a goal of the development team. The researchers not previously familiar with the software explored and examined it independently, so as to keep their opinions and observations unbiased. The research team conducted a series of user tests with six different testers. The researchers also utilized a set of game usability heuristics to conduct a Heuristic Evaluation (Pinelle et al., 2008). The research team primed the developers with a preliminary findings report, then sent a full usability report to the project's mailing list, both via email.

The published documents resulted in a flurry of responses, with 53 replies related to the original email. The final report, which was also delivered to the project's wiki, had one developer take it on themselves to add a comment field to each issue, as a means of facilitating discussion. The community took to this actively and worked to find solutions to the problems uncovered. The research team took part in all the discussions, keeping open dialogue between usability experts and developers. Continued expansion of the wiki related to issues found in UKKOSS 5 has been seen as much as a year after the research team stopped working on the project. On top of that several members remained in the project as developers, ensuring that many of the proposed changes are seen in the form of working code.

4.2.6 UKKOSS 6

UKKOSS 6 was run as a continuation of UKKOSS 5's work. It was conducted on the assumption that a project run with the groundwork already laid would be able to operate with fewer resources. As such, UKKOSS 6 took a less participative approach to its study. The researchers also assumed that users with different skill levels would evaluate the game differently in user tests. For example, a new player might have enjoyed how well a tutorial prepared them for the game to come, while a player familiar with the rogue-like genre could have been more distracted when that same tutorial pulled them out of the immersive experience and disrupted the flow of the game.

The software studied by UKKOSS 6 was the same rogue-like game that studied was by UKKOSS 5 a year earlier. It still had about the same size and scope, and two members of UKKOSS 5 were still in the project as developers, as one more joined the team based on his contributions to the project and prowess in the game itself.

The research team, since it was not as active a part of the development community, spent an increased amount of its time on conducting tests. They used Heuristic Evaluations based on both Pinelle (et al., 2008) and Desurvire (et al., 2004). Since the group was interested in how the user's skill level affected testing results, pre-interviews were conducted concurrently with the user tests, and three levels of users were identified. Post-interviews were also employed to provide context for the data collected by the researchers. Particular interest was paid to those who reported no prior experience with games in the rogue-like genre, since no complete neophytes were included in the UKKOSS 5 testing. In all, six people tested the game under the supervision of UKKOSS 6. Almost all communication between the research team and the development team was conducted through the usability champion, who stayed on as a developer from UKKOSS 5.

After the results were submitted, there were no direct developer comments on the specific results of UKKOSS 6, but they did state that they had read the findings and that the quality of the reports was professional and that the issues raised were valid.
Additionally, a wiki contributor took the time to reformat the submitted .PDF results into a proper wiki entry.

4.2.7 UKKOSS 7

Like UKKOSS 6, UKKOSS 7 was also run as a continuation of UKKOSS 5’s work. It was, similarly, conducted on the assumption that a project run with the groundwork already laid would be able to operate with fewer resources. Though UKKOSS 7 also took a less participative approach to its study, it also took a more focused one. The researchers assumed that users with lower skill levels would need more attention, so they focused on the game’s tutorial.

The software studied by UKKOSS 7 was the same rogue-like game that was studied by UKKOSS 5 a year earlier, and UKKOSS 6 in the same time period. Unsurprisingly, it had about the same size and scope, and two members of UKKOSS 5 were still participating in the project as developers.

The research team focused its efforts on making sure the tutorial of the game had the most attention, as the tutorial produces the first impressions new players will have of the software. The researchers conducted Cognitive Walkthroughs and User Tests, to evaluate the state of the software. Like UKKOSS 6, pre- and post-interviews were conducted to ascertain user levels and context. Also, as with UKKOSS 6, almost all communication between the research team and the development team was conducted through the usability champion (Schaffer, 2004), who stayed on as a developer from UKKOSS 5.

As the results were submitted with those of UKKOSS 6, there were similarly no direct developer comments. The usability activities did seem to be valued, they just did not create the same novel experience for the developers that was seen in UKKOSS 5.

4.2.8 UKKOSS 8

UKKOSS 8 attempted to make a return to the participative approach taken in UKKOSS 5, while operating within the same project. This change was made on the assumption that the factor missing from UKKOSS 6 and 7 was direct contribution to the software.

Again, the software studied by the research team was the same rogue-like game that was the subject of UKKOSS 5 through 7. No significant details about the project changed between studies.

Aspects of the study were all agreed upon with the development team, using the usability champion as a go-between for all community involvement. That method proved to be a problem for the research team, since there was a communications bottleneck created by that type of set-up. Mis-communications caused delays in testing tasks, and the usability champion further delayed reports, citing a need for alterations to be made in order for the reports in order to be applicable to the direction he wanted for the development of the game.

The specific results of this study have yet to be reported, as the study concluded after the publication draft of Rajanen et al’s (2012) paper was finalized.
5. Findings

Details of UKKOSS 10's usability testing research were presented in detail in an internal Test Report. An unedited copy of this report has been included in the Appendix. After those findings were established, the results of UKKOSS 10 were displayed in comparison to each of its predecessors, UKKOSS 1 through 8. No implications have been drawn at this time, so as to allow the reader a chance to examine the finding for his or her self.

5.1 UKKOSS 10's Analysis of todoyu

This section is going to focus on the information gained from the process of conducting Usability Testing. The specific details of UKKOSS 10 –its usability test report– are included in the Appendix for those interested in the evaluation of todoyu.

The researchers' actions and intentions can be best displayed in the form of the project's roadmap. This can also be used as a reference later, when and where the project deviated from its intentions. Please note that week numbers reference Finnish calendar weeks, and the project started on the third week of the year.

**Phase I: Project planning**
- Week 3: Kick off, selection of OSS project, project wiki pages up
- Week 4: Selecting OSS project (project selected), todoyu server up
- Week 5: Gathering information about todoyu
- Week 6: Gathering information about todoyu, existing expertise report, project plan and risk analysis
- Week 7: Existing expertise report, project plan, risk analysis, preparation for 1st Steering Group meeting

**Phase II: Usability test planning, testing and evaluation**
- Week 8: Creating a test plan, recruitment of test participants
- Week 9: Creating a test plan, usability testing within project team
- Week 10: Finalizing test plan, pilot test of the system
- Week 11: Pilot testing, improvement of test scenarios and tasks
- Week 12: Test sessions 1 and 2, transcription of test videos, analysing test results
- Week 13: Test sessions 3 and 4, transcription of test videos, analysing test results

**Phase III: Implementation design and implementations**
- Week 14: Design and implementation of usability improvements
- Week 15: Design and implementation of usability improvements, preparation for 2nd Steering Group meeting
- Week 16: Testing and evaluation of implementations
- Week 17: Testing and evaluation of implementations, communicating with the community
- Week 18: Evaluation of project teams contributions, making needed changes, communicating with the community

**Phase IV: Evaluation, feedback and finalizing the project**
- Week 19: Creating of project final report
- Week 20: Creating of project final report, preparation for 3rd Steering Group meeting
- Week 21: Finalize project
- Week 22: Finalize project
Using the email logs and meeting minutes, one can follow the actual progress of the UKKOSS 10 team through its entire process. That original source material provides a vast amount of raw data. In all, 87 pages of emails, 17 sets of meeting minutes, 37 pages of testing transcripts, and final testing reports were created as a part of this study. From that data, this paper compiled the following highlights, limiting the results to items relevant to the open source context and comparable to the findings of previous UKKOSS projects.

It is important to note that the author of this thesis collected this data as a participant/observer of the project. He was recruited by the researchers as an external usability specialist.

UKKOSS 10 team members went into the project aware of the previous UKKOSS projects. With the resources available, and a software developer on their research team, they felt they should try for as participative an approach as possible. Before selecting software to examine, the research team compiled literature on usability testing and open source software in addition to reading a selection of journal articles and conference papers, from Mikko Rajanen et al., relevant to the continuation of his work.

As a result of what was read, the UKKOSS 10 team went in with a number of assumptions. They knew they wanted a medium-sized active project with a well paced development cycle. They were also fairly sure they would, as in previous studies, be using Cognitive Walkthrough, Heuristic Evaluation, and User Tests. It was clear that the research team would need to be able to establish a relationship with the developers, so the presence of an active community around the software was added to the list of desirable project traits.

The project selection process began very loosely, with members bringing in ideas from wherever they found them.

In the end, three projects were brought to the table for consideration. That meeting was the author's first formal encounter with the project team, and he spent the meeting learning as much as he could about the three projects. He rejected the first project on account of its scope. It was far too large a project with a food chain that led to a known multi-national corporation. Thus, the project already had a vast human-computer interaction team, and the UKKOSS 10 group would have had little to offer and would have been able to make no independent decisions.

In considering the two projects remaining, it was noted that a section of the developers’ forums for the todoyu project management platform was asking for help with the user interface. This request pushed todoyu to the front, since all other characteristics of the two projects seemed comparable. Todoyu, itself, is a browser based open source software project with an array of modules and add-ons, some of which are commercial. According to todoyu's website, these are the key features of the todoyu project management software:

- Projects and tasks: manage multiple projects in one instance, keep track of your deadlines, assign responsibilities to your team members.
- Project management: always know what is happening and will happen in your projects, control deadlines, make sure you deliver on-time.
- Track time on a task base. With the billing module you can directly send out invoices based on your time sheets.
Report: Create powerful reports based on your projects, tracked time, bills, working types and many more. All calculations are done in real time. There is no easier way to keep track of your daily work!

Calendar: share dates for meetings, reminders, milestones, birthdays and many more. The calendar module reminds you by email or by a simple pop-up.

Manage your contacts: persons and companies.

Communicate with your customers with the help of the powerful commenting system.

Share: Comments, ideas, files, contacts and much more.

Focus on the essential: Running several projects parallel? Use the powerful filter functionality to focus on the essential. todoyu helps you to easily keep track of all the different deadlines, tasks and comments.

Open Source, BSD license

Web based

(todoyu.com, 2013)

Snowflake Productions oversees the development of the todoyu software and markets its commercial offerings.

Since the methods and tools for the usability tests had been mostly pre-decided, the main decisions to be made by the UKKOSS 10 research team concerned order and priority. Based on what had been seen in earlier iterations of UKKOSS, the research team wanted to let themselves be known to Snowflake and the todoyu developers as early as possible. The testing plans and research methodology were made as transparent as possible. Snowflake was informed of all major decisions the research team made and how they would affect the timetable. Snowflake in turn offered as much support as it could, including offering to supply members of its user base as testers. Wanting people who were new to the software, the UKKOSS team thanked the developers, but chose to select test users from the local community.

The testing of todoyu went fairly smoothly. A Cognitive Walkthrough was conducted to establish how users might be using the software and what kinds of tasks they would need to complete. From that, Heuristic Evaluations were conducted, identifying 13 usability issues, some with sub-parts.

The research team focused on interface issues during the Heuristic Evaluation, and found problems ranging from icon size to sub-menus which were difficult to locate. Five User Tests were conducted in a usability laboratory at the University of Oulu. Originally, four tests were planned, but the video data for one of the tests became corrupted when it was transferred from the lab’s computers, so a fifth was scheduled.

Each user was asked to test the system on eleven different tasks. The tasks ranged in difficulty and length. The tasks took participants 25 seconds or more to complete, with tasks being called off if they took more than ten minutes. Twelve usability problems were discovered through the testing, and each prompted suggestions for how to change the system or its interface.
Although the next version of the software had not gone live at the time of this publication, UKKOSS 10 is being labeled a success on the basis of the feedback the research team received from Snowflake Productions. Communications with the todoyu developer community never took off, but there was regular, quality communication in both directions between UKKOSS 10 and Snowflake. The research team was able to start work just as todoyu 2.3 was released, and version 2.4 had a stated goal of improved human-computer interaction and usability.

During the project, Snowflake wrote an article and posted it to their blog, praising UKKOSS 10 and discussing how excited they were to be getting professional testing done for their next release. “The Timing of this project for snowflake and todoyu is just perfect. The planning for the next Release is already in progress and since UI and Usability will be the main topics of 2.4, the Oulu Usability Project fits perfect into this planning” (todoyu.com, 2013).

Unfortunately, the back-end of todoyu was too complex for the research team to simply pick up and start developing, so implementing the suggestions for change falls on the todoyu community. Because of this, it is yet to be seen how much the usability testing impacts todoyu 2.4. Still, Snowflake has expressed interest in having usability testing included from the ground up in its next product, and remains positive in its attitude towards the continued use of usability activities in todoyu. Given those decisions, it is clear that UKKOSS 10 has had positive impact.

5.2 UKKOSS 10 Compared to Its Predecessors

Each of the UKKOSS projects did things in a slightly different manner. The most obvious difference was in the type of software that was being tested. The choice as to whether to be participatory of consultative ranked second. On the heels of that decision was one regarding the timing of the introduction of usability to the developers. Next were differences in the usability activities that were tried. Finally, the overall reception of the research group differed among the projects. Each of these factors will be looked at in relation to the choices made by UKKOSS 10.

Different types of software were tested and studied by each research team in the series of UKKOSS projects. UKKOSS 1 and 4 examined media center software with medium-sized development teams. UKKOSS 2, 5, 6, 7, and 8 all looked at games using slightly smaller but still mid-sized development teams. UKKOSS 3 was an outlier in that it examined professional 3D content creation software with a core-developer group larger than the whole development teams of any of the other pieces of software. UKKOSS 10 chose a project with a mix of these characteristics; its choice was professional software like UKKOSS 3, except with a less specialized use, that of project management. In the project selected by UKKOSS 10, the development team was mid-sized.

The approach taken by the various UKKOSS research teams was either consultative or participative. UKKOSS 1 and 4 chose to do their work as consultants, while the remaining research teams chose to take more or less participatory roles. The degree of participation varied throughout. UKKOSS 5 ended with two of the members staying on as full developers. UKKOSS 2 members took active suggestions from the developers as to what parts needed work and helped make design-level decisions in the form of suggested mock-ups. UKKOSS 3 was only active in that its members were trying to promote their team in the community; however, that community was so large that they got little attention. UKKOSS 6 and 7 were only participative through the usability champion, who did all of the community work for them. UKKOSS 8 barely managed to
<table>
<thead>
<tr>
<th></th>
<th>Software</th>
<th>Participatory /Consultative</th>
<th>Communication</th>
<th>Priming</th>
<th>Introduction</th>
<th>Usability activities</th>
<th>Results reporting</th>
<th>Reception</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Media-center</td>
<td>Consultative</td>
<td>None</td>
<td>No</td>
<td>At the end of testing</td>
<td>Cognitive Walkthrough Heuristic Evaluation</td>
<td>Report posted in discussion forum</td>
<td>Negative</td>
</tr>
<tr>
<td>2</td>
<td>Game</td>
<td>Participatory but treated as Consultative</td>
<td>Bottom-up</td>
<td>No</td>
<td>Upon arrival</td>
<td>Partial Cognitive Walkthrough Custom Heuristic Evaluation</td>
<td>Recommendations posted on the development wiki</td>
<td>Positive but limited</td>
</tr>
<tr>
<td>3</td>
<td>3D models</td>
<td>Participatory</td>
<td>Bottom-up</td>
<td>No</td>
<td>After identifying key players</td>
<td>Test results rejected</td>
<td>Report posted to development website and findings posted to community forum, IRC and community news</td>
<td>Negative but limited</td>
</tr>
<tr>
<td>4</td>
<td>Media-center</td>
<td>Consultative</td>
<td>None</td>
<td>No</td>
<td>At the end of testing</td>
<td>Cognitive Walkthrough Heuristic Evaluation</td>
<td>Report posted in discussion forum</td>
<td>Negative</td>
</tr>
<tr>
<td>5</td>
<td>Game</td>
<td>Participatory</td>
<td>Bottom-up</td>
<td>Yes</td>
<td>Before arrival</td>
<td>Partial Cognitive Walkthrough Custom Heuristic Evaluation</td>
<td>Emailed the project mailing list with initial findings then followed up with a full report</td>
<td>Positive</td>
</tr>
<tr>
<td>6</td>
<td>Game</td>
<td>Partly participatory</td>
<td>Bottom-up via usability champion</td>
<td>Yes</td>
<td>Long before arrival</td>
<td>Partial Cognitive Walkthrough Custom Heuristic Evaluation</td>
<td>Results submitted to the usability champion as a .PDF report</td>
<td>Positive but limited</td>
</tr>
<tr>
<td>7</td>
<td>Game</td>
<td>Partly participatory</td>
<td>Bottom-up via usability champion</td>
<td>Yes</td>
<td>Long before arrival</td>
<td>Partial Cognitive Walkthrough Custom Heuristic Evaluation</td>
<td>Results submitted to the usability champion as a .PDF report</td>
<td>Positive but limited</td>
</tr>
<tr>
<td>8</td>
<td>Game</td>
<td>Attempt at Participatory</td>
<td>Bottom-up via usability champion</td>
<td>Yes</td>
<td>Long before arrival</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>9</td>
<td>Project Manager</td>
<td>Participatory</td>
<td>Top-down via community manager</td>
<td>Yes</td>
<td>Before arrival</td>
<td>Cognitive Walkthrough Heuristic Evaluation</td>
<td>Periodic Skype updates and a report to the community manager</td>
<td>Positive</td>
</tr>
</tbody>
</table>
communicate properly with the usability champion and appeared to be participative in name only. UKKOSS 10 took an approach slightly different from any of these. They were active up though the design level, like UKKOSS 2, but the primary communication channel was between the project lead from UKKOSS 10 and the Community Manager from Snowflake Productions, with the work to be done by each team being directed from the top. The original goal was to include some development work in UKKOSS 10, assuring that the findings made their way into the next version of the project management platform. The latter task, however, proved to be too large for the resources available and Snowflake seemed capable of handling it on its own.

Several different tactics were identified when examining the point at which the research teams chose to announce their intentions to test their chosen software. Again UKKOSS 1 and 4 chose a different tack from the other teams; here, they waited until the project's end to let the developers know they existed, handing over a report of usability issues found and suggestions for correcting those issues. UKKOSS 3 took a more cautious route; the research team conducted a background investigation so that they would know the software and key players in its development, and then, when the time came to advance to usability activities, they announced who they were and what they had joined the community to do. In reaction to UKKOSS 1, UKKOSS 2 announced at the very start who they were and what they were there to study. The software project for UKKOSS 5 had a member of the research team on the development team. With this arrangement, the person who was on both teams was able to announce within the development project that usability was a priority and then quickly supply a team - the researchers- to conduct the testing. Following after UKKOSS 5, on the same software project, UKKOSS 6 through 8 continued research on a software project that had been tested and seen positive results from prior studies. Similarly, UKKOSS 10 did not have to introduce usability testing to anyone. Snowflake and the todoyu development community were looking for a user interface tester, so the UKKOSS 10 research group was able to offer an expanded version of that service, only needing to announce their research goals.

When conducting usability activities, all of the UKKOSS projects undertook User Testing. There was a small amount of variation in the usability inspection methods chosen to supplement the User Tests. Both Cognitive Walkthrough and Heuristic Evaluation were carried out by UKKOSS 1 and 4. Similarly, UKKOSS 2 used both methods, although the Cognitive Walkthrough was only done for the additional purpose of testing a level editor, and the Heuristic Evaluation was an alternate version created specifically for game evaluations. UKKOSS 3 organized usability inspections, but the results were found incompatible with the range and scope of the software project being tested. UKKOSS 5 and 6 also used gaming adaptations of the Heuristic Evaluation method, and UKKOSS 7 inspected the game's tutorial with a Cognitive Walkthrough. No mention was made in the conference papers about whether UKKOSS 8 utilized any usability inspections. Some changes were made during UKKOSS 10. While both Cognitive Walkthrough and Heuristic Evaluation were planned and conducted, the Cognitive Walkthrough was not deemed to have sufficient findings regarding usability issues, so it was just used by the research team to guide in the creation of later tests.

There was quite a range in how the results of the UKKOSS research teams were received. As with all the previous topics, UKKOSS 1 and 4 were the same in this regard; they were both almost completely ignored by the developers. The reception of UKKOSS 2 was a bit of a mix; many of their ideas were implemented, but only those that were thought important by the developers. UKKOSS 3 was even less of a success, in that it was heard, but not respected, and as soon as the researchers stopped posting, all of their work dried up. UKKOSS 5 stood out as having an amazing reception, and its efforts were still being built upon by the developers over a year after the study ended. In the shadow of UKKOSS 5, UKKOSS 6 and 7 quietly made contributions to the
game, but had very little impact on the development community. The final verdict on the impact of UKKOSS 8 is not yet known, but it seems likely that they only contributed a few game patches and only have a usability champion, who is frustrated by miscommunication, to remember them. UKKOSS 10 made an unusual impact, in that it did not stir up the developer community, but it was used as a banner by the leaders of the software project. Like UKKOSS 8, the next release is not available, so it has yet to be seen if the changes suggested by the usability tests were taken to heart or not. Yet unlike UKKOSS 8, UKKOSS 10 was praised openly on Snowflake Productions’ blog. Additionally, Snowflake has promised to include even more usability activity in their forthcoming products.
6. Discussion

After collecting and analyzing all pertinent data, several deductions can be made. In this section, the research question will be simply answered. Then, its implications will be discussed in as much detail as possible.

6.1 Research Question

“What actions lead to successes and failures when conducting usability testing on open source projects?”

Not unsurprisingly, the actual act of conducting usability activities looks very much like it would in closed source software (Nielsen, 1994). The primary differences appear to occur with how the process gets started and how it is wrapped up. That is to say, special attention must be paid to introducing the notion of usability into a project, consistent with Rajanen et al. (2011), and the results of the testing need to be formatted and delivered based on that specific community’s practices. Which tests are chosen and how they are conducted have an impact on the quality of the results, but beyond that, seems to have no importance to the software developers. Which is to say, it is extremely important, but not because of the open source context.

It would appear that size and structure of a software project should be taken into consideration, but that there is no clear evidence at this point as to whether large projects make usability testing harder or simply that they need to be handled in a different fashion. The different development structures of software that were tested and studied by individual research teams appear to have a moderating effect. Specific structure types do not directly determine which projects have been labeled successes or failures, but they coincide with the effectiveness of certain methods. In the UKKOSS series, usability activities were less successful when applied to projects with more complicated development structures, perhaps because testing results did not propagate throughout the community.

The clearest factor to success seems to be taking a participative approach to joining the software community (Rajanen, 2011). Projects operated in a purely consultative role appear more likely to meet with a failure to impact the software or its developers.

The point in the process at which a research team announces its intentions is a factor that affects usability outcomes. Yet, the interactions seem parabolic in nature. Projects that wait until the end to let the developers know they exist are a clear non-entity, but teams who are known about a year or two in advance have a similar, if less potent, anonymity. Those teams which let their actions be known, at or just after the start of their study, meet with moderate success. The most successful teams, however, have their intentions known just before they arrive on the project.

Usability inspection methods chosen to supplement the User Tests do not drastically affect the success of the usability activities. In the research series, UKKOSS 3 organized usability inspections whose results were found incompatible with the software project being tested, and UKKOSS 10 decided not to publish the results of their cognitive walkthrough. On the other hand, the teams from UKKOSS 2, 5, 6, 7,
and 8, which tailored their inspections to gaming, were able to use the usability findings from their inspections. Therefore, the use of methods that match the software tested appears to make for a more efficient use of time and resources.

6.2 Implications

As a means to provide concrete suggestions to usability professionals, a methodology of sorts could be inferred from the results:

Software Development Style → Communication style → Priming → Tests Types → Results.

In a sentence: The software development style of the project to be tested should determine the communications style used when priming the developers with the knowledge of what usability testing you will be conducting and how you will deliver the results.

When planning to include usability activities as part of an open source software project, the first set of actions needs to be to identify the structure and hierarchy of that project. Given the variety of ideologies that can be found under the free/libre/open source umbrella, not all projects can be treated the same. Each project will have its own reason for being open source, and that could range from the desire to keep communication free, to a belief that everyone should be able to use the software as they want, to a simple desire to develop the software with the help of volunteers. A professional production team might welcome the aid of usability experts who can add value to their project, whereas someone who independently created software, which can be used by anyone as they see fit, might see usability results as criticism of the hard work they have done. Almost everything the usability professional does with a project should be related to back to the development style under which the software is being created, not just to the nature of the software being created.

These findings on project hierarchy are consistent with those found in the literature. Crowston & Howison (2006) detailed the variety found in open source software projects, admitting surprise in their discussion section in regard to how much variation there was. The same phenomenon was examined again by Valverde & Solé (2007) in their paper *Self-organization versus hierarchy in open-source social networks*.

Hierarchy played a major role in several of the UKKOSS projects. Most notably, it was a blockade to progress in UKKOSS 3. The research team approached the project the same way as they would a small open source community, and found themselves adrift in layers of bureaucracy. Inversely, UKKOSS 1 and 4 handed a formal report to a group of open source developers with no context; this may have been taken as a statement of inadequacy on the part of the developers.

Communications with the software developers should be conducted with an understanding of the type of developers in the forefront. Conversations with the developer community can be started in one of two ways. The first is “bottom-up,” where the usability professional establishes validity and builds its case among the forums and channels of the developer community. The second is “top-down,” where communications are directed toward high level decision makers within the project. This decision will be based on several things, most importantly the development structure. The more formal and structured a hierarchy there is in a project, the harder it will be to have individual ideas rise to the top. Therefore, it is important in those cases to find someone with decision making power to put the ideas into the project for you. Likewise, a looser structure will likely have a more democratic approach to feature
implementation, so it works best to gain popularity in the community and use that to gain momentum for usability-testing.

The idea of communicating with different group structures in different ways has long been in the literature. Originally, it was a topic of interest to business and management, as seen in Krackhardt’s (1993) Harvard Business review article. Later, it made the transition to related disciplines and can be found in information systems and open source software literature in Crowston & Howison’s (2005), *The social structure of free and open source software development*.

In UKKOSS, communications were conducted in several different ways. Most of the research teams chose to involve themselves from the bottom up. After being unable to secure a direct line with top management, UKKOSS 3 set up a dedicated team for communicating in the forums and on the IRC channel, in attempting to convey the legitimacy of usability activities to the community from the bottom up. UKKOSS 10 was a salient example of taking the top-down approach. The researchers in that group had access and were therefore able to communicate directly with the *todoyu* community manager, who championed their cause for them. In some cases, quality usability testing was conducted, but communications were lacking, as demonstrated by the experiences of UKKOSS 6, 7, and 8.

When it is time to get started, a good way to initiate communication is through priming. The idea behind priming is simple: Get the developers to believe that it was their idea to bring usability to the project. The usability professionals should appear after the concept of usability is already on the table or even after the value of usability has already been accepted. This can be accomplished in several ways. The project developers can genuinely come to the realization that they need usability on their own. A usability team member can join the project ahead of the rest of the team in order to champion the use of usability from within, or the usability specialists can convince a champion within the project to rally the cause for them. Whatever method is used, the community must feel that they need usability before the usability team is brought in to conduct tests.

This type of social priming stems from the literature of social and behavioral psychology. After being primed with a course of action, “People sometimes find themselves going along with the group reflexively—without much thought and without knowing why,” according to Epley & Gilovich (1999). This phenomenon has been seen in corporate settings where companies looking to do business with each other have more success when they are primed as being favorable business partners (Wang, 2007).

Priming was seen in UKKOSS in the projects that met with the most success. In UKKOSS 5, a member of the usability research team was already a known developer for the software. As such, he was able to advise bringing in usability experts from an existing position in the community. UKKOSS 10 had usability primed for them by the community manager, who noted that it was a skill they were lacking and advertised the need on their development website. This is, of course, ideal, but it is not a situation that can be intentionally recreated for research. In the case of UKKOSS 3, an attempt was made to prime the development community, but the research team moved on to conducting their usability research before their members had established validity within that community. Inserting your own usability champion should be seen as a more long-term approach and should not be relied upon when timing is critical.

The testing itself should never be routine. No matter how many times a usability testing professional has conducted testing, he or she should still start from the ground up. Methods and testing tasks should be created based on the specific project and its needs.
Industry standards can work as starting points, but will typically need to be adjusted to fit the current environment.

Choices as to what tests to run are everywhere in the literature and textbooks. It has been fairly clearly established that every software engineering task is unique (Holzinger, 2005), so the tests will have to be selected accordingly. Still, for the sake of completeness—and for the benefit of new usability experts—the point should still be included here.

UKKOS' testing results provided some useful information. All the research teams conducted user tests, and all of them created their own tasks to be tested, but each team chose slightly different usability inspection methods to accompany the user testing. Choices regarding which tests to conduct, in each UKKOS project, were as interesting in the projects where the chosen methods did not work, as they were in the ones where they did. While it was good to see the UKKOS 2 team create its own hybridization of different gaming heuristics for their Heuristic Evaluations, an examination of UKKOS 3 and 10, which had to discard data due to incorrectly selecting their inspection tools, teaches the reader more. Had UKKOS 10 spent more time considering options, the research team may have had a richer set of findings to report back to Snowflake Productions.

A research team's findings can be released in many forms and should reflect the project they are evaluating. A usability report may be appropriate for a hierarchical project, yet inappropriate for a loose organization that is working from a wiki page. It is important to formulate results in a manner that reflects the community they are being created for. This may be a report, a blog post, forum discussions, a software patch, or one of any number of other reporting methods.

The reasoning for using different styles of result reporting comes from an unexpected place in the literature. This notion came from the ideas presented in Chartrand's (2005) *The role of conscious awareness in consumer behavior*. In this case, since the results of usability tests are the product that usability specialist want to see adopted, the core idea for this originated in consumer psychology. The same idea has already been applied to other topics, like translation (Gengshen, 2003), where -like an open source team- there is a specific culture to be taken into account.

Reporting of UKKOS results was accomplished in several different ways. The majority of the UKKOS teams created reports. Although a typical way to report research projects, the reports were not always well received. In some cases, as with UKKOS 6 and 7, the reports were re-written by community members to fit with their wiki-style communications. In others, such as UKKOS 1 and 4, the reports were acknowledged, then ignored. In the case of UKKOS 10, however, reports were expected, as the community manager needed to know what his best course of action was in order to proceed. UKKOS 3 marked a different type of reporting; its findings were disseminated through community forums and news sites, and despite the fact that the team did not receive the amount of attention they were looking for, some developers took the suggestions for change into account when working on the project. Therefore, the findings did have some effect on the usability of the software.

By following a methodology, such as this one, usability experts are able to to address the divide between Usability Testing and Open Source Software Development. The usability specialists can participate as part of the collective, and open source developers can incorporate answers to usability problems while they themselves move on to solve problems that they find interesting (Raymond, 1999). In the process of approaching projects on their own terms and formatting results to fit the individual project's style,
usability activities can exist without the gap seen previously (Bødker, 2007; Zhao, 2005; Moghaddam, et al., 2011).
7. Conclusions

As open source software has long been the domain of in-house production and personal use programs, many publicly available pieces of software are being created, in whole or in part, with open source projects. As this happens, open source coders, who are accustomed to creating software for themselves, are facing the need to create a product that is expected to be usable by people who are not technology aficionados.

7.1 Contribution

This thesis examined the process of implementing usability in open source software projects. In doing so, it compared the tasks that were conducted in a series of usability testing projects to explore what did and did not lead to success. As a result of this examination, a loose methodology was proposed for conducting usability activities on open source software projects.

The usability testing community benefits from this research in the form of a new potential methodology to use when operating with open source software projects. Even though the methodology is still untested, it can provide food for thought and promote usability within the discipline. Once it has been verified, it could provide a framework for properly including usability activities in the development cycle.

Information systems research has been given a body of case studies it can use to examine how the process of usability testing plays out under various conditions within open source development. It can find additional utility from the examinations of those case studies in comparison to each other.

Human-computer interaction research should be able to utilize both the methodology and the examination of the case studies. The methodology should assist in creating suitable user interfaces, while the case studies will provide additional data to the body of knowledge regarding human interactions.

In conclusion, usability testing can be carried out within open source development projects without the dissonance that has historically accompanied it. By adjusting communication style to fit the software development style of a project, usability experts can effectively prime the developers, supplying them with better knowledge and understanding of which usability tests will be conducting and why. As a consequence, when the usability-test results are delivered, they are in context and easier to understand, which in turn makes them easier to implement.

7.2 Limitations

No project is without limitations, and this one is no exception. The first thing to be noted is the fact that all of the research teams in this study consisted of students. The stance of this paper is that such teams are not in fact a limitation in terms of data produced. They operate on a slower schedule than more experienced usability professionals would, but each student had completed a series of usability courses and was overseen by a human-computer interaction researcher to assure rigor was
maintained and quality results were delivered to the software development teams of the products being tested.

Of greater importance was the fact that the research teams were not able to be involved in the software projects from the beginning. Unfortunately, this is not feasible, as open source software projects take time to gather community and develop into full scale projects. As a result, despite knowing this limitation, it was deemed an acceptable loss to join existing teams and attempt to instill a usability culture after the fact.

The final limitation for this study came from the fact that not all of the researchers had the same goals. This was important to the UKKOSS 10 portion of the study, wherein most of the researchers were focused on the Usability Testing as an end in itself. The fact that UKKOSS 10's team was also working on a university course precluded picking and choosing other aspects of the research case to tailor, since they had specific tools they were expecting to use for the course. Because of this limitation, a comparison to all preceding UKKOSS projects was selected in exclusion of directly studying todayu's development empirically.

### 7.3 Recommendations for Future Research

This study readily lends itself to continuation in future research. Obviously, there are many variables that have not yet been included in the study, and with the exception of UKKOSS 1 and 4, the study results have not yet been replicated. Taking UKKOSS 10 as an example, a second study using the same assumptions and a similar case could be conducted to assure that the findings here are consistent with what is seen in the field. Working with a company-sponsored open source project, a top-down approach to communication could be taken to introduce usability into the project. In this way, further research could validate the top-down approach or any other variable a research team found interesting.

As another avenue for study, a research team could include usability testing in a project starting from the design phase and continuing throughout the process. Due to the new relationship with Snowflake Productions, a team could start from the ground floor on the development of todayu's mobile application. Since the groundwork has already been laid, there is room for a team to continue the partnership and study the benefits of including usability activities in the starting phases of software development.

Along the same lines as the previous avenue of study, it would be of great benefit to look specifically at methods of project entry. The how, when, and why of usability's insertion into projects would contribute to the usability experts effectiveness and their perceived value in the community at large. By the same token, each of the steps of the proposed methodology could be taken out and studied in isolation.

Of greatest interest to the author of this thesis, the loose method outlined in the implications section could be tightened and empirically tested. The links between Development Style, Communication Style, Priming, Test Choices and Presentation of Results could all be analyzed in relation to each other. With additional validation, the methodology could be formalized and introduced to the research community.
References


Appendix - UKKOSS 10 Usability Test Report

Usability testing (812671S)
Test report
todoyu

University of Oulu
Information Processing Science
Tuomo Ruottinen
Deepak K.C.
Hana Asfaw
Matthew Barth
Siqi Fu
16.05.2013
Abstract

TodoYu is an open source web-based project management system and it was chosen as the object system for usability testing. TodoYu enables the user to create and manage projects, track the projects progress and communicate with the members of the project team and the projects customer. The software includes features that allow management of tasks, deadlines and reports. The goal of testing is to test and evaluate the usability of the core features of todoYu. Moreover, the aim of usability testing is to contribute the results of the usability activities for the OSS developers and to implement the possible changes based on those usability test results. The testing team consists of five members, one pilot and five actual test participants from outside of the usability team.

TodoYu’s usability evaluation is divided into two groups: heuristic evaluation group and cognitive walkthrough evaluation group. The data gathered from internal testing group is used for testing scenarios and tasks. Usability evaluation and internal usability testing was done by the project team members individually. The actual usability testing had five test sessions and it was done in the usability testing laboratory. We found different usability issues through usability test sessions and heuristic evaluation and we also provided recommendations and designs.
Contents

Abstract .......................................................................................................................................................................................................................... 2
Contents ......................................................................................................................................................................................................................... 3
1. Introduction .............................................................................................................................................................................................................. 4
   1.1 Description of object of usability testing .......................................................................................................................... 4
   1.2 Goals of usability testing .......................................................................................................................................................... 5
   1.3 General usability questions ...................................................................................................................................................... 5
   1.4 Test methods ................................................................................................................................................................................. 5
   1.5 Choosing and recruiting test persons ................................................................................................................................. 6
   1.6 Definition of test scenarios and test tasks .......................................................................................................................... 6
      1.6.1 Scenario #1 Admin with system administrative activities ......................................................................................... 6
      1.6.2 Scenario #2 Project manager with project management activities ................................................................................... 6
      1.6.3 Scenario #3 Project member with regular project task activities .................................................................................... 7
   1.7 Test sessions, pre and post interviews ................................................................................................................................. 7
   1.8 Execution of expert evaluation for the test object .................................................................................................................. 8
   1.9 Execution of a pilot test ............................................................................................................................................................ 8
   1.10 Execution timetable .............................................................................................................................................................. 8
2. Execution of usability testing ........................................................................................................................................................................ 10
   2.1 Test sessions .............................................................................................................................................................................. 10
   2.2 Test recordings ......................................................................................................................................................................... 11
3. Analysis and interpretation of findings .................................................................................................................................................. 12
   3.1 Analysis of the collected materials ......................................................................................................................................... 12
   3.2 Interpretation of usability findings .......................................................................................................................................... 12
      3.2.1 Quantitative results ........................................................................................................................................................... 12
      3.2.2 Qualitative results ............................................................................................................................................................ 14
4. Usability findings and recommendations .......................................................................................................................................................... 16
   4.1 Forgot Password Wizard ......................................................................................................................................................... 17
   4.2 Front page after login .............................................................................................................................................................. 18
   4.3 Search Box ............................................................................................................................................................................... 19
   4.4 Top Navigation Bar .............................................................................................................................................................. 20
   4.5 Contacts Management ....................................................................................................................................................... 21
   4.6 New Task Button ............................................................................................................................................................. 22
   4.7 Assistant Wizard ............................................................................................................................................................ 23
   4.8 Projects ................................................................................................................................................................................. 24
   4.9 Main menu .......................................................................................................................................................................... 25
   4.10 Planning ........................................................................................................................................................................... 26
   4.11 Calendar .......................................................................................................................................................................... 27
   4.12 New user ......................................................................................................................................................................... 28
5. Lessons learnt ..................................................................................................................................................................................... 29
Appendix ........................................................................................................................................................................................................... 30
1. Introduction

This report emphasizes the usability test results and findings of todoyu project management system. This chapter describes the system, the goal of usability testing more in detail and the test cases. Chapter two describes how the usability testing was executed. Chapter three focuses on the analysis and interpretation of the findings. Chapter four describes usability findings and recommendations. In chapter five we described what we learned from the whole usability testing process.

1.1 Description of object of usability testing

Usability testing was done to find and record any usability issues in todoyu project management system interfaces. The goal of usability testing is to support the tasks of the system administrator, project manager, project team members and project customers. The testing was done with the combination of exploratory testing and usability evaluation methods. The testing process was divided into two phases, within the project team an internal testing phase and with test participants an external testing phase. The system was tested based on three user roles; system admin, project manager and project team member. Testing of the functionality of the system started from the login screen. The core features of the system were tested which include reporting of project schedule, tasks and results, communication tools and other parts of the system. The test results were used to plan and design user interface redesigns to fix known and found usability issues.

User experience, effectiveness and learnability were the key attributes used as usability criteria. The system should be as easy, intuitive and learnable as possible for the intended user, and give sufficient feedback. The system should also be as effective as possible in the sense that the user gets his/her project management tasks done in reasonable amount of time. The user interface should not include, for example, useless elements, but have only those functions visible that are relevant for that particular part of the interface. The figure below shows the portal of todoyu system. This user interface displayed after the users logged in by using their username and password.

![Figure 1. todoyu system](image-url)
1.2 Goals of usability testing

The main goal of the usability testing was to collect both quantitative and qualitative data from new users, how they use todoyu to perform certain tasks when using the system for the first time. We expect the result of our usability testing to be used by the todoyu community for the further development of the todoyu system. Every team member and five other participants representing different age group, backgrounds and qualifications, tested the core functions of the system.

1.3 General usability questions

The following usability questions were used as criteria to evaluate test participants in the testing sessions. The post-test session interview replies by each participant was another criteria used in the process.

- Concepts: Do the users understand the concept of the todoyu system?
- Discoverability: Is the functionality visible to the users?
- Accessibility: Are the users able to access all the features of todoyu?
- Behavior: Do the users notice and understand the popup notification?
- Content: Is there additional content the users expect to see?
- Customization: Are there additional customization options the users might want?
- Desirability: Do the users find todoyu system more appealing over other similar systems?
- Look and feel: What are the users’ reactions to the visual look and feel of todoyu?
- Security: Is the todoyu secure to use?
- Interaction: How do the users interact with different todoyu features?
- Comparison: Is there functionalities provided by other management systems that the users would desire to see in todoyu?

1.4 Test methods

The internal (project team) testing was done by using two expert evaluation methods, heuristic evaluation and cognitive walkthrough. The project team was divided into two groups; the first group was using heuristic evaluation to check all of the usability aspects and reporting them accordingly. The second group was using cognitive walkthrough of parts of the system to assess the usability of the system and to record testing tasks for the participant testing phase.

The external test phase was done as exploratory testing. We used different data collection methods for usability testing, which include observation, interviews, video analysis and thinking-aloud. Test participants tested the system by doing through a list of predefined tasks for a testing session. First the facilitator briefed the test participant about the system and asked pre-interview questions before a testing session started, to get basic information about their background. Post interview question were asked after the testing session, to gather their user experiences and observations during the test session.
1.5 Choosing and recruiting test persons

There were five test participants in the actual testing phase with ICT educational and working background. The participants had prior experience in project work, so they were familiar with project tasks and also had some understanding in project management. The target groups of the system are project managers of small to medium sized projects. The role of a project manager is to create a project and manage project team members, project schedule and tasks, communication and reporting. The participants had good computing skills. We named the test participants as P1, P2, P3, P4 and P5 to protect the privacy of the participants. P2, P3 and P5 had prior work experience in ICT sector. P1 and P4 are ICT students.

The ages of the test participants were between 20-30 years. We recruited the test participants from the University of Oulu through personal relationships existing with the testing team. The agreement with the test participants was done orally. We offered small rewards for their valuable contributions for our usability testing, like coffee and chocolate after they completed the test session. None of the participants had ever heard of the todoyu project management tool before.

1.6 Definition of test scenarios and test tasks

The test scenarios are provided as role play scenario task lists for each todoyu’s predefined roles: system admin, project manager and project team member. The role play scenarios motivated the test participants to do the testing and to immerse them into the testing situation. The length of a testing session took between 40 to 60 minutes depending on the testing task and the test participant. List of tasks was provided step by step in a printed form. The list below shows three different scenarios and the list of tasks.

1.6.1 Scenario #1 Admin with system administrative activities

Task A: The ICT Company, Rapid Testing, uses todoyu as their project and task management system for their daily project activities, and you are the administrator of Rapid Testing's todoyu system. To start your job, add Rapid Testing and their partner, Adelphi, as companies in the system.

Task B: You need to staff your projects, so add Mr. John Smith as a customer, working for Adelphi, and add Ms. Sirpa Doe as a project member, at Rapid Testing. Give them access to the todoyu system.

Task C: These people want to use the calendar. Give Mr. Smith the right to use the calendar's functionalities, and give for Ms. Doe the right to use the calendar area.

1.6.2 Scenario #2 Project manager with project management activities

Task D: You are now working as the project manager for one of Rapid Testing's OSS projects, and currently, you have several different projects and tasks to manage. Create a project called “OSS,” with the simple description of "First project." This project is in progress, since it started on 14.01.2012 and doesn't end until 15.05.2013. Adelphi is the customer for this project, with John Smith acting as the contact, and Sirpa Doe is going to lead the project from Rapid Testing's end.
**Task E:** You wanted to give a specific task to one of the project members. Add a “usability testing” task to the project. This task starts on 14.01.2012 and has to be to the customer by 15.05.2013, assign the testing task to Sirpa. It is expected that the task will take Sirpa 100 work hours to complete.

**Task F:** You want to track the time spent on specific tasks in order to send bills to customers. Start time tracking.

**Task G:** To stay on track, you want to schedule a “weekly meeting” event in the calendar every Wednesday, starting date on 13.03.2013 and an ending on 29.05.2013, from 15:00 -16:00 hours.

**1.6.3 Scenario #3 Project member with regular project task activities**

**Task H:** You want to change your password in todayu. Change your password from “PassworD123” to “ABc123”.

**Task I:** You have completed your OSS testing report, and you want to add the test report file to the usability testing task interface. Add a test report file to OSS project.

**Task J:** You want to inform your co-workers about the usability testing task report. Write comment in the usability-testing task that says, “I have done the introduction and objective sections of the usability testing report.” Link the report to this note.

**Task K:** You have many different tasks from different companies, and you wanted to search Inferno Studios to find out its contact person. Use the global search to find a company starting with “Infer,” and use the results to view its contact person.

**1.7 Test sessions, pre and post interviews**

The tests in internal testing phase were done partially at home for pretesting, and the actual testing was done in the University of Oulu’s usability testing laboratory. The tests in participant testing phase were done completely in the usability testing lab.

The testing lab has a laptop for usability testing, equipped with two web cameras to record the user (test participant) activities. There is an observation computer on the other side of the testing lab behind a screen with three monitors to record the screen of the testing laptop and to observe the testing sessions. The testing laptop has a screen recording software (Camtasia) installed to record videos of the testing process.

The test moderator and test observer did usability testing observations. The cameras attached to the testing laptop recorded the test sessions and the test videos were saved on the observation computer. A short interview of the test participants was done before the testing sessions to gather background information about the test participants, and a longer interview was held after the testing sessions to assess the experiences of the test participants.

Our usability testing team consists of five members. During the testing process each member of the team was involved in different activities to fulfill their duties and responsibilities. There were three persons who participated in one usability test session which are the data logger, the test moderator and the test observer. The role of the test moderator was to interview and instruct the test participants and manage the testing sessions in general. The test observer was seated quietly on the other side of the testing lab to observe the activity of the test participant taking notes.
The test participants were welcomed appropriately to the test sessions by the testing team. The facilitator or the moderator was responsible to instruct and support the test persons during test sessions. The facilitator provided the task lists for the test participants before session begin. Moreover, the facilitator encouraged the test person during the test session. During a test session there was no interaction between the other part of the testing team and the test participants.

We used interview and observation data gathering techniques to support our test results. We observed the test participants activities while interacting with the system during the test session. Semi-structured interview as a questionnaire was conducted before the test session to understand the background information of the test participants. Another post-test session interview was conducted after the test session to understand the test person’s impressions and experience after using the todoyu. The pre-test session questionnaire and post-test session interviews are attached in the appendix.

1.8 Execution of expert evaluation for the test object

Two expert evaluation methods were used, Nielsen’s heuristic evaluation and cognitive walkthrough. The project team was divided into two groups, Tuomo and Matthew did the cognitive walkthrough and Deepak, Hana and Siqi did the heuristic evaluation. The heuristic evaluation was divided in three parts, from admin perspective, from project manager and project team member perspectives. During the evaluation we concentrated more on the icon and button handling of the system.

1.9 Execution of a pilot test

Before conducting the pilot testing and the actual usability test sessions, we wanted to try the system and the task lists within our usability testing team. We gathered in the usability testing lab and two of our members were responsible to perform the task lists and other members were observing and taking notes. We had difficulties to finish tasks in the middle of scenario two. Therefore, we could not complete all of the task lists, because when a project was created there was an error message on the rate set field that would not let us to move to the next step. We decided to try it out of the lab and discuss the problem with the todoyu developers. Moreover, some of the tasks were leading to some extent and we decided to improve them. The pilot test was conducted on March 10, 2013. During the testing session, there was very little difficulty. However, we changed some of the procedures and filled the required information before the actual testing.

Execution of the pilot test helped us to understand how the test scenarios and procedures are well formulated. The pilot test participant was someone who does not have any ICT background because we wanted to know how the tasks are understandable by a non-ICT experienced users. We used the three test scenarios for the pilot test that is explained in section 1.6.

1.10 Execution timetable

We conducted five actual usability test sessions and one pilot test. The actual test conducted between March 21 and March 27, 2013. Before the actual test we conducted pilot test to understand how the scenarios and task lists works as expected. The table below shows the summary of the testing schedule for each participant.
<table>
<thead>
<tr>
<th>Participant</th>
<th>Testing Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot</td>
<td>Sunday, March 10, 2013</td>
<td>02:00 PM - 03:00 PM</td>
</tr>
<tr>
<td>P1</td>
<td>Thursday, March 21, 2013</td>
<td>12:00 AM - 02:00 PM</td>
</tr>
<tr>
<td>P2</td>
<td>Friday, March 22, 2013</td>
<td>04:00 PM - 05:00 PM</td>
</tr>
<tr>
<td>P3</td>
<td>Tuesday, March 26, 2013</td>
<td>10:00 AM - 11:00 AM</td>
</tr>
<tr>
<td>P4</td>
<td>Wednesday, March 27, 2013</td>
<td>04:00 PM - 05:00 PM</td>
</tr>
<tr>
<td>P5</td>
<td>Wednesday, March 27, 2013</td>
<td>05:00 PM - 06:00 PM</td>
</tr>
</tbody>
</table>

Table 1: Test session execution timetable
2. Execution of usability testing

The tests in internal testing phase were done partially at home for pretesting, and the actual testing was done in the University of Oulu’s usability testing lab. The testing lab has a laptop for usability testing, equipped with two web cameras to record the user (test participant) activities. There was an observation computer on the other side of the testing lab behind a screen with three monitors that recorded the screen of the testing laptop and observed the testing sessions. The testing laptop has a screen recording software (Camtasia) installed to record videos of the testing process.

2.1 Test sessions

The main ideas of the task lists emerged from the quick start manual of todoyu which is found in the todoyu website. The tasks were designed in order to accomplish the main functionalities of the system by the administrator, project manager and project team member. Formulating the tasks was an advantage for us because it helped us to understand the system more in depth. The task lists were revised and edited to make the usability test participants more comfortable while they are testing the system. The table below summarizes demographic information of the pilot test and the actual test participants.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Age</th>
<th>Gender</th>
<th>Occupation</th>
<th>Previous experience with project management and project management tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot</td>
<td>30</td>
<td>F</td>
<td>Student</td>
<td>Some project management experience with group work</td>
</tr>
<tr>
<td>P1</td>
<td>26</td>
<td>M</td>
<td>Student</td>
<td>Some project management experience with “Microsoft project”</td>
</tr>
<tr>
<td>P2</td>
<td>23</td>
<td>M</td>
<td>Web designer</td>
<td>Little experience with “Trello.com”</td>
</tr>
<tr>
<td>P3</td>
<td>26</td>
<td>M</td>
<td>Game designer</td>
<td>Some experience with project management tools but did not use any project management tool</td>
</tr>
<tr>
<td>P4</td>
<td>25</td>
<td>M</td>
<td>Student</td>
<td>Some experience with “targetprocess”</td>
</tr>
<tr>
<td>P5</td>
<td>27</td>
<td>M</td>
<td>Entrepreneur</td>
<td>Some project management experience</td>
</tr>
</tbody>
</table>

Table 2: Test participant demographic information
2.2 Test recordings

The videos recorded during the test sessions were saved to the usability testing team personal external hard disks and were shared to all testing team members by using FileSender service. The pre-pilot test and the actual test sessions were recorded. Unfortunately, the video recorded on the usability test session 3 was corrupted and we could not recover the data. All the recorded videos were transcribed into a document. The video has good voice and picture quality, however in some part of the video it was difficult to understand what the test participants were trying to say. The recorded video and the participants’ action were easy to see clearly because Camtasia has a zoom-in feature built-in.
3. Analysis and interpretation of findings

We collected all the necessary materials including interview and observation notes, videos, and video transcriptions to structure and categorize the problems found in the usability test sessions. We used qualitative and quantitative analysis methods. Quantitative analysis includes time and number of paths taken to complete each task, number of errors encountered in attempt to complete a task and the success rates for each task. Qualitative analysis includes test participant's verbal comments made during think-aloud protocol, facial expressions and results from the post-test session interview.

3.1 Analysis of the collected materials

We analyzed the interaction of the participants with the object system. We transcribed all of the videos that include the activities and speech of the test participants. P1 transcription is attached in the appendices as a sample. The recorded videos analyzed individually and in a group. One of our usability team members made 20 minutes video that is gathered from all usability test session videos, the video shows how test participants manage to do a task on the system (task E). We gathered to watch the video with our supervisor to find out usability issues as a group. We observed that all of them were doing the task in a different way.

We watched the first participants’ video together and analyzed the video. We set a sort of criteria to analyze the rest of usability testing videos individually. We eliminated talks held between tasks and we counted number of errors encountered in each task. We calculated the average time to complete a task and total time to finish one test session. Calculating time of completion helped us to understand how easy a task was and how fast the participants discovered the functionalities.

3.2 Interpretation of usability findings

3.2.1 Quantitative results

The major errors encountered in tasks A, B, D and E were very basic ones. In majority of the cases participants did not care much about the password requirements and did not provide the information for the compulsory fields. Participant one encountered 1 error in task A, 4 error in task B and 4 error in task D. Participant two encountered 1 error in task A, 3 errors in task B, 2 errors in task D and 3 errors in task 3. Participant four encountered 7 errors in task A, 3 errors in task B, 2 errors in task C and 3 errors in task E. Participant five encountered 4 errors in task, 3 errors in task D and 2 errors in task E. We did not record how many errors participant three encountered because the video was corrupted. The combined usability testing videos are available at the following url:


The table below shows the approximate time to complete a task, the average time to complete a task and the total time to finish a usability testing session.
<table>
<thead>
<tr>
<th>Minutes and seconds/task</th>
<th>Pilot</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>Average time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task A</td>
<td>02:00</td>
<td>04:00</td>
<td>03:50</td>
<td>05:00</td>
<td>07:25</td>
<td>03:30</td>
<td>04:21</td>
</tr>
<tr>
<td>Task B</td>
<td>05:00</td>
<td>06:30</td>
<td>06:20</td>
<td>07:00</td>
<td>08:35</td>
<td>05:10</td>
<td>06:33</td>
</tr>
<tr>
<td>Task C</td>
<td>12:00</td>
<td>02:70</td>
<td>06:10</td>
<td>10:00</td>
<td>03:40</td>
<td>04:00</td>
<td>06:36</td>
</tr>
<tr>
<td>Task D</td>
<td>04:30</td>
<td>04:00</td>
<td>08:30</td>
<td>04:00</td>
<td>04:60</td>
<td>04:35</td>
<td>04:93</td>
</tr>
<tr>
<td>Task E</td>
<td>03:00</td>
<td>02:15</td>
<td>08:90</td>
<td>04:00</td>
<td>03:75</td>
<td>02:55</td>
<td>04:06</td>
</tr>
<tr>
<td>Task F</td>
<td>00:25</td>
<td>01:00</td>
<td>01:50</td>
<td>01:00</td>
<td>01:30</td>
<td>00:65</td>
<td>00:95</td>
</tr>
<tr>
<td>Task G</td>
<td>04:30</td>
<td>05:15</td>
<td>06:50</td>
<td>10:00</td>
<td>03:00</td>
<td>07:55</td>
<td>06:08</td>
</tr>
<tr>
<td>Task H</td>
<td>02:00</td>
<td>00:30</td>
<td>00:85</td>
<td>01:00</td>
<td>00:90</td>
<td>00:40</td>
<td>00:91</td>
</tr>
<tr>
<td>Task I</td>
<td>01:00</td>
<td>01:00</td>
<td>01:05</td>
<td>01:00</td>
<td>02:00</td>
<td>04:10</td>
<td>01:69</td>
</tr>
<tr>
<td>Task J</td>
<td>01:30</td>
<td>01:00</td>
<td>02:06</td>
<td>03:00</td>
<td>00:50</td>
<td>01:00</td>
<td>01:48</td>
</tr>
<tr>
<td>Task K</td>
<td>01:00</td>
<td>00:40</td>
<td>01:20</td>
<td>01:00</td>
<td>01:30</td>
<td>00:50</td>
<td>00:90</td>
</tr>
<tr>
<td>Total</td>
<td>36:15</td>
<td>28:00</td>
<td>46:16</td>
<td>47:00</td>
<td>36:35</td>
<td>33:50</td>
<td>37:90</td>
</tr>
</tbody>
</table>

Table 3: Task completion times
3.2.2 Qualitative results

All of the test participants were hovering around before they found the create icon. However, after they first found the create icon, it was easier to find other elements and whenever they had difficulties to find some functionality they would always check the create icon. Participants clicked the save button repeatedly because they did not notice the popup message. We observed that participants were not frustrated when they performed tasks.

Participant comments and our observations during the test sessions

- One of the participants felt that the create icon was hidden.
- Three of the usability test participants noticed the “add new person” label written two times.
- When participants were asked to give rights and roles for using calendar functionality and area, they were expecting to be able to give rights for a single person. One of the participant thought that he could manage rights and roles from the person’s profile.
- Most of the participants found the label “assets” confusing.
- Participants tried to select estimated workload time that is more than “99.55” hours from the scroll bar, but they did not manage to select more than that. Therefore, they wrote 100 hours manually.
- Participants did not find add task easily from the create project field, they first clicked other buttons first before they clicked “Menu”.
- Some of the participants did not complete creating event and they found the calendar confusing with lot of functionalities. Two of the participants experienced calendar bugs.
- Only one of the participants used the filtering search menu properly but other participants used the magnifying icon for searching.

Participant comments on the post interview questions

We summarized the post interview as follows and we put interesting comments of the participants in quotations.

Pilot: She thinks that it works quite well to see how companies are interconnected and she liked it. She was fairly comfortable when she was completing the tasks. She likes the interconnectivity the most because it’s handy to link a lot of people from multiple companies. She recommends todayu to medium to larger size companies that do a lot of business with small companies that do a lot outsourced business, because it’s a really good way to keep a database of everyone you are working with, and who's working exactly with whom. If she used the system daily she probably gets used to it within a week.

“I think that it would make sense if creating a new thing were accessible from multiple places rather than just the taskbar”.

P1: It was a bit messy when he used it at first time, but after he used it, it was quite nice. He thinks it’s quite ok for bigger project management. He likes the interface because it was easier to find necessary information. He thinks one may become a proficient user if the system is used for one week at least. He said that it would be nice if the footer part was improved and if some kind of help is added.
“I think those tasks you have are pretty good, in all around experience I can use it quite easily”.

P2: He was uncomfortable on a couple of situations for the first time but it may be easier if he used it more. He likes the billing functionality and that the customers are included in the system. He likes the icons but he thought the labeling was not clear and it was not consistent. It will not take him more than a week to be a proficient user of the system. He thinks it’s better for company’s employees in a bit larger scale projects like software projects that would last like a year.

“It is a complex system in a way so far for me to become usable. I think it would need a kind of major facelift”.

P3: He thinks that the user interface is fiddly but he was comfortable doing tasks. He likes the dropdown menus. He could be able to use the system well approximately within a week. He recommended that the font size could be a bit bigger.

P4: He was comfortable after he did some tasks but it was confusing at the beginning. He likes the search filter because he does not have to write the whole sentence and he can search in multiple conditions.

P5: He said the system was quite usable and he was comfortable when he was doing tasks. He said that at the beginning the elements were not easy to find but after that he likes the elements because they are fixed and stable. The rights and roles were confusing and he commented that it would be nice if it’s improved and the extensions look complicated. It may take him one week or less to get used to the system.
4. Usability findings and recommendations

This chapter documents a heuristic evaluation of todoyu, an open source project management tool developed by snowflake productions GmbH. The heuristic evaluation was conducted by the team UKKOS10, a group of 5 students studying in Oulu University in The Department of Information Processing Science. The goals of this usability testing (heuristic evaluation) is to evaluate the usability of todoyu using a set of heuristics and disclose a list of issues to developer’s attention. The findings along with the possible solutions mentioned in the document will guide the developers for future changes to improve the interface of todoyu.

The team used Nielsen’s ten usability heuristics (Nielsen, 1994) to evaluate different aspects of todoyu. Each team member along with 5 other participants went through a set of tasks multiple times to record usability issues and the corresponding heuristic violations that occurred. The issues are then rated in terms of severity. The severity scale used is as below:

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Don’t agree that this is a usability problem</td>
</tr>
<tr>
<td>1</td>
<td>Cosmetic problem</td>
</tr>
<tr>
<td>2</td>
<td>Minor usability problem</td>
</tr>
<tr>
<td>3</td>
<td>Major usability problem; important to fix</td>
</tr>
<tr>
<td>4</td>
<td>Usability catastrophe; imperative to fix</td>
</tr>
</tbody>
</table>

Nielsen’s ten usability heuristics used are as follows:

1. Visibility of system status
2. Match between system and the real world
3. User control & freedom
4. Consistency & standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose and recover from errors
10. Help and documentation
4.1 Forgot Password Wizard

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forgotten Password Wizard</td>
<td>2+4+7</td>
<td>3</td>
<td>BRIEF DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The forgotten password wizard allows resetting password by entering only username.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIKELY DIFFICULTIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Forgetting username is very likely and in such situations, users cannot reset their password until they contact the administrator.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC CONTEXT/LOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location: Password Reset Wizard Page</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSSIBLE SOLUTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Add functionality to reset username and password by email id with which it is possible to reset password either by providing user id or email id. (Figure 1 in the Appendix)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HEURISTIC RATIONALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2) Match between the system and the real world: Users do not have normal password resetting feature as most other systems do.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4) Consistency &amp; standards: To reset username either by providing username or email id is a standard process which is missing in the system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7) Flexibility &amp; efficiency of use: The current function does not cater to all types of users.</td>
</tr>
</tbody>
</table>
### PROBLEM

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
</table>
| Front page after login | 7+8 | 4 | **BRIEF DESCRIPTION**

The front page after logging in as an administrator should display a page with administrative tasks rather than To Do’s. It is usually the last page that the user has selected in the previous logon: My Filter, To Do’s, Feedback or Events.

**LIKELY DIFFICULTIES**

It takes a while for a novice user to figure out what an admin possibly can do.

**SPECIFIC CONTEXT/LOCATION**

Location: Front page after logging in

**POSSIBLE SOLUTIONS**

The current front page should be replaced with a page that contains all major tasks an administrator can do. (Figure 2 in the Appendix)

**HEURISTIC RATIONALE**

7) Flexibility & efficiency of use: The admin features are scattered in the navigation bar.

8) Aesthetic & minimalist design: The front page shown currently is of less relevance or rarely needed. Administrator would not like to see To Do’s after his logon; he would rather prefer functions like User Management, Project Management, and Billing in the front page.
### 4.3 Search Box

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Search Box</td>
<td>7</td>
<td>1</td>
<td>BRIEF DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>There should be a search box on the top to search relevant items, when on projects tab it should automatically search only projects, when on Contacts tab should automatically search only contacts unless the user supplies a filter to search.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIKELY DIFFICULTIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The current search feature is on top represented by a magnifying glass. For a novice user, it is difficult to find the search function, as he/she would normally look for a search box.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC CONTEXT/LOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location: Every pages after logging in</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSSIBLE SOLUTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The search box should be visible in all pages and the magnifying glass on top can be replaced. (Figure 3 in the Appendix)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HEURISTIC RATIONALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7) Flexibility &amp; efficiency of use: The search box only appears after clicking the magnifying glass.</td>
</tr>
</tbody>
</table>
### 4.4 Top Navigation Bar

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Top Navigation Bar</td>
<td>7</td>
<td>3</td>
<td>BRIEF DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The top navigation bar for a beginner is extremely difficult to understand. The icons are not clear. The sub-icons under the major icons are repeated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIKELY DIFFICULTIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>It is difficult to find out the exact task behind the icons. One needs to click on the icon and check if the function he/she is looking for is present within the mentioned icon.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC CONTEXT/LOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location: Top Menu icons in all pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSSIBLE SOLUTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The menu icons should be customized and added text so as to give clear ideas to users about the icons. More icons with clear information can be added (optional). (Figure 4 in the Appendix)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HEURISTIC RATIONALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7) Flexibility &amp; efficiency of use: It takes a while to figure out the function hidden within an icon.</td>
</tr>
</tbody>
</table>
## 4.5 Contacts Management

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTIC S</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Contacts Management</td>
<td>3+6+7</td>
<td>3</td>
<td>BRIEF DESCRIPTION</td>
</tr>
</tbody>
</table>

On clicking the current Contacts Menu item, the list of persons and companies is displayed under corresponding tabs, which allows edition and deletion of single contact information. It lacks the feature to add new contacts, and remove several contacts together.

**LIKELY DIFFICULTIES**

The user is not able to add a new contact and delete several users when on Contacts Tab.

**SPECIFIC CONTEXT/LOCATION**

Location: Contacts page when clicking contact menu on the top

**POSSIBLE SOLUTIONS**

To add check boxes in one column in the table and an option to delete them together by making a button at the end of the table that can allow several options of editing or deleting to the selected user or companies. To add new contact feature in the sub menu after clicking contacts. (Figure 5 in the Appendix)

**HEURISTIC RATIONALE**

3) User control & freedom: Users are restricted to add new user and delete several users together.

6) Recognition rather than recall: Being able to see the main links associated with a certain feature reduces memory load on the user.

7) Flexibility and efficiency of use: To add a new contact, one has to go on the top icon bar, the feature is not available under submenu in Contacts.
### 4.6 New Task Button

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
</table>
| 6. New Task Button | 4 | 0 | BRIEF DESCRIPTION
On clicking the plus sign on the top menu, the New Task item is repeated twice.

LIKELY DIFFICULTIES
It confuses people and makes them wonder why there are two New Task options when both of them function the same way.

SPECIFIC CONTEXT/LOCATION
Location: Every pages after logging in

POSSIBLE SOLUTIONS
One of the New Task buttons should be removed.

HEURISTIC RATIONALE
4) Consistency & standards: Two New Tasks make people wonder if they mean the same thing or are different.
## 4.7 Assistant Wizard

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTIC S</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Assistant Wizard</td>
<td>3+7+8</td>
<td>3</td>
<td>BRIEF DESCRIPTION</td>
</tr>
</tbody>
</table>

The assistant wizard that appears on the middle of the screen is useful. However, the current location of the wizard window and it appearing every time you navigate to a new page is disturbing to the normal workflow. On closing assistant window, it appears when you refresh the page or click some other links within the system.

### LIKELY DIFFICULTIES

The wizard hides everything and makes normal working process difficult. It is not possible to get help from the wizard and work at the same time. After disabling the wizard, there is no way to get it back if user wishes to take help from the wizard.

### SPECIFIC CONTEXT/LOCATION

Location: Every page after logging in until the Assistant wizard is disabled

### POSSIBLE SOLUTIONS

Its location should be changed to the bottom so the rest of the components are visible. An expandable and collapsible div can be used to hide or unhide the assistant wizard with options to close, disable and enable the wizard. (Appendix)

### HEURISTIC RATIONALE

3) User control & freedom: No support for enabling the assistant wizard after it is disabled.

7) Flexibility & efficiency of use: The assistant wizard, despite being helpful, is annoying and disturbing the normal workflow.

8) Aesthetic and minimalist design: The location of the assistant wizard is irrelevant.
## 4.8 Projects

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>8. Project tab</td>
<td>6+7+8</td>
<td>3</td>
<td>BRIEF DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>On clicking the project tab, it displays the last project that was selected. It also contains tabs for other three projects to select.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIKELY DIFFICULTIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>It is confusing as you can only see 3 projects and there is already a project being selected.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC CONTEXT/LOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location: On the Project page</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSSIBLE SOLUTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>It should display the list of projects and on selecting the one it then should provide the current features that can be applied to a project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HEURISTIC RATIONALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6) Recognition rather than recall: One has to remember that the sub menu under projects displays only 3 projects. The rest of the projects are on left hand side and can be searched.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7) Project navigation should be easy and displayed as a list rather than a tab that contains only 3 projects.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8) Aesthetic and minimalist design: On clicking project tab, one assumes a list of projects to be displayed rather than only 3 projects.</td>
</tr>
</tbody>
</table>
### 4.9 Main menu

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>9. Main Menu</td>
<td>2</td>
<td>2</td>
<td>BRIEF DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The menu items in the main menu are not clear. The selected words and poorly organized menu items confuse users.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIKELY DIFFICULTIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Difficult to find the appropriate item one is looking for.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC CONTEXT/LOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location: Main menu in every page</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSSIBLE SOLUTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>To properly categorize the menu and use proper words that are matching the current world situation. Planning can be replaced with Events. Portal does not make any sense. It should be replaced with Quick link that is customizable with the possibility for a user to add or remove items he/she is using frequently.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HEURISTIC RATIONALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2) Match between system and the real world: Words used in describing menus are not common or familiar ones.</td>
</tr>
</tbody>
</table>
### 4.10 Planning

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. Planning</td>
<td>2+6+7</td>
<td>3</td>
<td>BRIEF DESCRIPTION</td>
</tr>
</tbody>
</table>

The planning tab in the main menu allows viewing calendar in daily, weekly or monthly format.

**LIKELY DIFFICULTIES**

The format to view the calendar is not expected under the planning tab. Users usually look for adding new event and customizing the added ones.

**SPECIFIC CONTEXT/LOCATION**

Location: Main menu in every page

**POSSIBLE SOLUTIONS**

Under planning the links should be changed to Add new Event & Edit New Event.

**HEURISTIC RATIONALE**

2) Match between system and the real world: The word used to describe event manager is not usual and the option a user normally sees under planning tab is not present.

6) Recognition rather than recall: The user has to remember to click + button in order to add any event.

7) Flexibility and efficiency of use: No accelerators like quick link to adding event are present currently.
4.11 Calendar

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>8</td>
<td>2</td>
<td>BRIEF DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The calendar has too many options to navigate between months and days which is normally not present in a calendar.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIKELY DIFFICULTIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Confuses users with too many options. They are not able to get to the right button and they need to keep on trying.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC CONTEXT/LOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location: Calendar under the planning tab</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSSIBLE SOLUTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>To eliminate the unnecessary buttons on the calendar and design it to be a normal calendar.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HEURISTIC RATIONALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8) Aesthetic and minimalist design: The current calendar system contains irrelevant or rarely needed buttons.</td>
</tr>
</tbody>
</table>
### 4.12 New user

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>HEURISTICS</th>
<th>SEVERITY</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>New User Form</td>
<td>2</td>
<td>2</td>
<td>The user adding with the form is a lengthy process with unclear fields.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIKELY DIFFICULTIES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>It causes difficulties when adding a new user to the system. The confusing fields make users unsure about what they are doing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC CONTEXT/LOCATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location: A form which appears on clicking New Person</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSSIBLE SOLUTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The unnecessary fields should be eliminated, proper terms and help should be used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HEURISTIC RATIONALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2) Match between system and the real world: Dummy user is not a familiar and commonly used when creating a new user. There are no force password change options for the user and no place to retype the password to confirm it.</td>
</tr>
</tbody>
</table>
5. Lessons learnt

The usability testing process went well to produce some interesting outcomes and recommendations for the todoyu community. Technical challenges were experienced during the testing process. The video files produced by the Camtasia software used were not compatible with other video editing applications or tools. We relied most of the time on the free version of Camtasia Studio. Exporting videos to AVI or other formats to make testing videos compatible with other video players lost the quality of videos. Editing and combining tasks from different sessions too impacted the quality of videos. We would recommend to look for another tool for the purpose of recording, or if possible, the existing application Camtasia should be configured to produce videos in the form that is compatible with various video players available.

There was good collaboration among usability testing team members and different types of expertise among us helped us in implementing testing sessions, setting up todoyu systems for the test participants, smooth documentation, reporting and analyzing.

The participant selection used in the process is not a convincing one. Majority of the test participants fall under the same age category and qualification that affected the output of the tests. There were no demographic variations either.

The whole process went quite well and was productive, however if we would have to do the testing again we would emphasize more on testing plan, proper time allocation for the whole testing process and participants selection process.
Appendix

1. Forgot Password Wizard

![Forgot Password Wizard](http://www.theeduzone.net/project2/frontpage.png)

2. Front page after logging in as administrator

![Front page after logging in as administrator](http://www.theeduzone.net/project2/frontpage.png)

Better Quality Image at: [http://www.theeduzone.net/project2/frontpage.png](http://www.theeduzone.net/project2/frontpage.png)
3. Search Box

4. Top Navigation Bar

Better Quality Image at: [http://www.theduzone.net/project2/topmenu.png](http://www.theduzone.net/project2/topmenu.png)
5. Contacts Management

![Contacts Management](http://www.theeduzone.net/project2/contacts.png)

6. Assistant Wizard

![Assistant Wizard](http://www.theeduzone.net/project2/assistant-suggestion-1.png)