

UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Informatics Curriculum

Erik Berendsen

Usability testing of e-shop software

Bachelor's Thesis (6 ECTP)

Supervisor(s): Dietmar Alfred Paul Kurt Pfahl

Tartu 2014

Usability testing of e-shop software

Abstract:

This paper researches usability testing methods and tries to derive an effective method for testing e-shop software. The derived method shall be used on e-shop software in order to determine if the suggested method could be used for usability testing on this type of software.

Keywords:

Usability, e-shop, software, testing

E-poe tarkvara kasutatavuse testimine

Lühikokkuvõte:

Selles bakalaureusetöös uuritakse kasutatavuse testimise meetodeid ning pakutakse välja e-poe tarkvara kasutatavuse hindamiseks meetod. Seda meetodit rakendatakse e-poe tarkvaral, peale mida saab hinnata selle meetodi päriselulist rakendust e-poe tarkvarade kasutatavuse hindamiseks.

Võtmesõnad:

Kasutatavus, e-pood, tarkvara, testimine

I. Table of Contents

1. Introduction	5
1.1 Previous and similar work	6
1.2 What are we going to evaluate?.....	6
1.3 Contributions	7
1.4 Overview	8
2 Modern usability evaluation methods	9
2.1 Usability inspection methods.....	9
2.2 Usability testing methods	10
2.3 User experience assessment.....	12
3 Description of evaluation method	13
3.1 Determining the objective of the evaluation.....	13
3.2 Which parts of the software will be tested?.....	13
3.3 Evaluation organization	14
4 Application of evaluation method	23
5 Results from evaluation.....	26
5.1 Results from usability testing	26
5.2 Lessons learned about the method.....	29
6 Summary	32
7 Kokkuvõte	33
Bibliography.....	Error! Bookmark not defined.
Appendix	36
I. Test result analysis	36
II. Video observations.....	39
1. Tester 1.....	39
2. Tester 2.....	41

3.	Tester 3.....	43
4.	Tester 4.....	44
5.	Tester 5.....	45
III.	Additional notes.....	47
IV.	Scenarios.....	48
V.	License	50

1. Introduction

A comprehensive evaluation is an assessment that covers the need, design, implementation, impact, and efficiency of a program or software [1]. There are many ways in which the data gathered from an evaluation can help the software developer team to rethink their product to better match the needs of the consumer. For example, if the evaluation results indicate that there is currently no need for the product in the market, then perhaps the core functionalities of the software need to be changed to better suit the needs. Or perhaps the product should be cast off. Due to the huge popularity in online shopping, more and more e-shop software are being developed. This naturally brings on a need amongst marketers and developers in a comprehensive usability evaluation of that type of software. The evaluation would help make decisions regarding further development and marketing. The goal of this thesis is to research modern evaluation methods and derive a method for evaluating e-shop software. Deriving it involves conducting a test with the method and determining the effectiveness of it.

Modern software testing covers a large array of aspects such as performance, security, maintainability, functionality etc. In this thesis, we are concerned about the usability of e-shop software. Usability assesses the quality of a user interface – how easy it is to use. It is defined by 5 quality components: learnability, efficiency, memorability, errors, and satisfaction [2]. In the recent years there has been a rising shift in the emphasis on user experience. User experience views a person's perceptions and responses that result from the use or anticipated use of a product [3]. Thus usability and user experience overlap in many ways but their main difference is that usability focuses on the functional part while user experience is concerned with the emotions stemmed from aspects of the product.

1.1 Previous and similar work

Usability testing is conducted widely on many different such as consumer electronics and various sorts of objects like TVs. In computer science, usability studies look at the ease of interaction between a human and a system. This means that there exists a large number of similar works. The work that was found to be most similar will be brought out. As for previous works, no studies of usability methods for e-shop software's could be found.

The most similar paper that was found is called "A user experience study of airline websites" [4]. As the name suggests, they were investigating the user experience of different airline websites. The aim of their study was to identify similar problems in three airline websites and provide recommendations. The similarity between that study and this thesis is that they were also testing a very specific type of web pages. However, the main difference is that they were not trying to evaluate the method that they used for the testing.

1.2 What are we going to evaluate?

The software we will be evaluating is called Kauplur. It is an easy to set up e-shop template software. The main purpose of the software is to make it easy for inexperienced, mainly Estonian, users to host a web shop in under an hour. It is built upon the web software Wordpress and an open source e-commerce plugin called WooCommerce. Wordpress uses themes to allow users to change the look and functionality of a WordPress website or installation without altering the information content or structure of the site. Kauplur uses a theme called Pinboard [5] which was created by Daniel Tara and modified by the development team.

Kauplur was created during the Software Project (MTAT.03.138) course by a team consisting of Erik Berendsen, Jaagup Viil, Rainer Viro and Kaarel Tõnisson. The core functionalities of the software were to make e-shop setup as simple as possible and to simplify the process of managing the e-shop. During the evaluation we will be focusing on the e-shops owner and customer interfaces. For the end user, shop management should be as easy and effortless as possible. During the evaluation we wish to see where and why users come across problems which affect the intended use. If we can see a trend where many people get stuck at the same place for the same reason then we can make adjustments to improve the software usability.

The target audience includes small business owners, entrepreneurs and individual vendors who wish to set up an e-store to sell their merchandise. For example a gardener who would like to sell his or her flowers online.

1.3 Contributions

Table 1 shows the potential beneficiaries of this thesis, as well as why this thesis could benefit them.

Beneficiary	What do they want to know?	How will they use the result?
Developers of the program	Where do people come across difficulties? Why do they come across difficulties? Is there a need for such software?	To make decisions about user experience modifications. To make decisions about further investment in the software.
People interested in e-shop evaluation methods	How to create an evaluation method for e-shop software?	In making decisions about possible evaluations of their own.
Clients	Is the software useful for them?	In making decisions about using the software

Table 1: Beneficiaries

1.4 Overview

In the first chapter an introduction about software evaluation shall be given. Also in the introduction the purpose of this thesis is provided as well as potential beneficiaries. In the second chapter of this thesis an overview of modern evaluation methods shall be given to provide a better understanding of the diversity of evaluation methods. The third chapter will be a detailed description of the suggested evaluation method for e-shop software derived from analysis of the modern methods and goals for the evaluation. The next step is to test the suggested evaluation method. The description of the application of the testing technique will be in the fourth chapter. In the fifth chapter, assessment of the evaluation method which was suggested in this thesis will be given. Finally, in the sixth and seventh chapter, the summaries – first in English and secondly in Estonian.

2 Modern usability evaluation methods

In this chapter we describe some of the methods used for conducting usability and user experience tests. Usability is a part of the overall user experience. Usability testing can be split into two major categories: usability inspection and usability testing. Usability inspection methods are all based on having evaluators inspect a user interface. Usability testing on the other hand focuses on having users evaluate the software. Each one has its advantages and weaknesses. It is up to the evaluation conductor to determine which method best suits the current situation at hand.

2.1 Usability inspection methods

One common usability inspection method is heuristic evaluation. Heuristics determine usability problems associated with the design of user interfaces. Heuristics are techniques for solving problems contrived from experience and intelligence. This approach takes a holistic view to identify problems. The most known heuristic is trial and error. Heuristic evaluations are usually conducted by an expert who reviews an interface against a set of guidelines or principles. There is no single set of heuristics but the most popular ones are Nielsen's heuristics. They are as follows: visibility of system status, match between system and the real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help users recognize, diagnose, and recover from errors, help and documentation [6][7]. According to Rolf Molich heuristic evaluations should be done prior to or in addition to user-testing, not instead of user-testing [6].

Another usability inspection method is cognitive walkthrough. This method works by analyzing how easily a new user can accomplish tasks within the software. The distinguishing factor about this inspection method is the fact that users prefer the hands-on approach instead of following instructions or reading a guide when learning new software. Each task is analyzed and the required effort to accomplish the task is specified. This effort can include other subtasks which have to be completed beforehand. Then the tasks are carried out while typically asking analytical questions about each task. After answering these questions, the expert can determine possible usability problems. There are usually four main questions asked [8]:

- Will the user try to achieve the effect that the subtask has?
- Will the user notice that the correct action is available?

- Will the user understand that the wanted subtask can be achieved by the action?
- Does the user get feedback?

The last usability inspection method looked at will be pluralistic walkthrough. This method involves users, developers and usability experts all participating in completing different tasks within the system or software. Because of interaction between different types of participants it is highly likely that many usability problems will be found. Due to the large amount of people involved, a large amount of resources are needed for this type of evaluation.

There are many more usability inspection methods such as heuristic estimation, feature inspection, standards inspection and consistency inspection, but because of the circumstantial criteria in which they should be used, they will not be covered in detail.

Usability inspection methods are generally considered to be cheaper than usability testing methods [8]. They always involve usability expert(s) evaluating the usability of software by comparing certain criteria. This means that the results of the testing can be partial due to the inevitably incomplete knowledge of the expert(s). Inspection methods are commonly used at early stages of development in order to look for problems before any real coding has been done. The software being evaluated in this thesis has been completed and because of the unavailability of a usability expert, these types of methods are not suited for this evaluation.

2.2 Usability testing methods

User-testing is another common software testing method. This way of testing involves testing the software on actual users to observe how a person perceived a system. This type of testing provides valuable feedback because it gives direct input on how actual end users will possibly use the software [9]. Usability testing is done in a controlled environment to determine the ease-of-use of the system being evaluated. This means that the users should have a product which they can use. The test conductor observes the process to try and pinpoint tardy or inoperative functionalities. These can be anything from badly placed buttons and unintuitive design to broken functions or links. Conducting a usability test involves having the user complete certain tasks using the software being tested, both of which should be provided by the test conductor. The tasks should be carefully chosen to be the most important and frequent tasks done using the product. Testing every task would be impractical due to limited time of the users and large

number of possible tasks. The goal of the tests is to observe how users function in a realistic setting performing tasks which are common in the product being tested. The techniques for gathering data during testing also vary. Widely used techniques include test monitoring, direct recording, think-aloud and eye movement tracking.

Hallway testing is a method where five to six random people who are not involved with the project do the testing. The fact that no trained testers are needed means that this method is comparably cheap. The people chosen should not be developers or engineers because their advanced knowledge of the product means that they already know how to accomplish given tasks, thereby missing ambiguities and false paths.

Expert review is a type of method where a usability expert is brought in to evaluate the usability of a piece of software. The expert uses a set of guidelines to measure basic usability criteria such as learnability, design etc. There are guidelines that have been developed, but the expert can create a new set if needed. Expert reviews can also be automated, these are called automated expert reviews.

Automated expert reviews are similar to expert reviews as they use guidelines to evaluate software's usability. The difference is that the tests are done automatically. The positive side of this type of method is that it is quick and consistent. On the negative side, the information provided might not be detailed enough.

Remote usability testing is, as the name suggests, a method for testing usability when the test conductor and users are separated. They are categorized by time – synchronous and asynchronous. Examples of synchronous remote usability testing are remote application sharing software and video conferencing. Nonsynchronous testing includes gathering data collected by logging the user's activities. The biggest advantage to this method is the wide range of different people it can cover. This is because to user can do the test from anywhere in the world.

2.3 User experience assessment

User experience measures how people feel about the item under evaluation. User experience assessment is non-trivial because user experience is subjective, depends on context and changes in time [10]. Because of this and limits of the scope of this research, user experience assessment will only be covered briefly. Usability testing is one part of user experience testing.

User experience assessment can be split into three categories – implicit, explicit and creative methods. Implicit methods try to find patterns in users' nonverbal actions such as face expression checking and eye tracking. Explicit methods try to make the user describe their emotions and thoughts. To do this, emotion assessment is used to evaluate a person's momentary reactions to an interface's user experience. Creative methods try to bring together the design team and the target audience's visions and ideas to create a rich user experience. There are also methods to determine how a person's emotions about a products user experience change over time, they are called longitudinal user experience assessment methods.

There are many different testing methods out there and each has its own positive and negative sides. When deciding on any of the methods it is important to take the context of your evaluation into consideration. Different methods may yield different results even if used on the same test group.

3 Description of evaluation method

3.1 Determining the objective of the evaluation

The purpose(s) of this evaluation is to improve the usability of the software under evaluation. In order to make software's user experience better, the strong and weak points must be determined. This will be the main question I will be trying to answer during the evaluation – Where are the strong and weak points in the user experience?

1. Which functionalities are hard to find?
2. Which activities are hard to complete?
3. How hard is it to learn?
4. Which functionalities are easy to use?

3.2 Which parts of the software will be tested?

The evaluation will cover the user experience using computers and not mobile or other devices. This is because the e-shop management is usually done using a computer and the limited timeframe inhibits testing both mobile and computer.

During this evaluation we will be most interested in the way users interact with the customer and owner interfaces of the e-shop software. We will be seeing how intuitive the interfaces are and how long does it take for the users to do certain tasks. The tasks will involve common and some not common activities.

Use cases

Now I will present the use cases which go over the system functionalities for both the e-shop owner and the customer. Most are taken from the functional requirements of the software development [11] though some are chosen by the evaluation conductor based on expert knowledge of the domain. The list of use cases is given below:

Owner

1. Choosing a preferred language
2. Changing the shop title

3. Creating/removing/updating posts
4. Changing the background of the website.
5. Changing the header and footer background color
6. Adding a new product
7. Adding a product category
8. Creating/removing/updating pages
9. Managing orders

Customer

10. Choosing a preferred language
11. Searching for products by name
12. Adding/removing a product to/from the shopping cart
13. Changing the amount of product in the shopping cart
14. Successfully placing an order
15. Commenting/liking products using Facebook

Based on these use cases it is possible to describe scenarios. The use cases used will be picked by the test conductor based on expert opinion. Both scenarios will be played out by every participant. First the shop owner scenario and right after that the customer scenario. These scenarios should be designed to help the evaluator recognize problematic usability areas. This means they should emulate real life scenarios in real life environments. The scenarios have to be designed to answer all the evaluation questions.

3.3 Evaluation organization

The test will involve conducting a qualitative systematic observation under controlled conditions. The testing will be performed in a one-on-one environment where the participant is actively involved in using the product and discussing their success and failures in real-time. This type of environment means that any organizational questions that the participant may have can be solved quickly as well as problems which may cause the participant to quit the test prematurely. However, this does not mean that the test conductor will in any way help the participant complete the scenarios. The test will be timed. Each test starts from a clean slate, meaning that after each test session the test environment will be reset to the state where it was before the session. This is

done by deleting the database and reimporting the backup database made of the state before the test. At the beginning of the test, a pre-test questionnaire is filled to establish basic information about the participant. The participant first play the role of the e-shop manager and secondly the customer role.

The test method that will be used in this paper is a combination of scenario testing, which is a type of hallway testing, and the think-aloud method. Scenario testing involves a hypothetical story used to help a tester think through a complex problem. The ideal scenario test has several characteristics, it is: based on a story, motivating, credible, complex use, easy to evaluate [12]. The think-aloud method is, as the name suggest, a method where the tester is encouraged to speak their thoughts out loud. It is used to evaluate a person's intentions and their actions.

The scenarios are presented as task-oriented use cases. Scenarios can help identify problems in a certain task and potential task completion times. Scenarios are good because of their low resource requirement and ability to generate context for evaluation studies. The only resources needed are people to test use cases on and an environment to test in. The think-aloud method will be used because it makes the observation process much easier while potentially inducing stronger emotional reactions from the participant. This is done to uncover the usability issues and emotions derived from the user experience. At the end of the test session the user will fill in a questionnaire about his/her experience using the software.

There should be a way to monitor the test so that the test conductor can keep track of the user's movements in the software. The monitoring should be remote. This is because we want users to be in a situation where they would be when using the software on their own. For later analysis it is vital that we record the think-aloud process because it is not reasonable to think that recalling everything said by the test participant from memory is possible. All this should ensure a more real life scenario and more accurate data. The software we will be using for monitoring the user end of the screen is called Camtasia Studio [13]. This software was chosen because of its screen and voice recording capabilities.

In order to gather additional information from the test participant, questionnaires will be conductor before and after the test. Online questionnaire software will be used because it helps save time and simplified the formatting process of creating a questionnaire. The questionnaires

will be conducted using an online questionnaire website called SurveyMonkey [14]. This software was used because it is free.

This approach was chosen because our focus is on usability and active participation in the use of the product will ensure that we get honest and constructive feedback. A one-on-one environment also makes sure that any questions can be answered right away making sure each session is successful in completing the test.

The steps for conducting the test are listed here in chronological order:

1. Pre-test questionnaire
2. Explaining the test method (think-aloud) to the participant
3. The computer screen and participants audio are recorded
4. Participant is given scenarios to play out
5. After the scenarios have been played out the user fills in a post-test questionnaire.

Completion of the questionnaire marks the end of the test.

Shop owner scenario

This scenario will cover the owner use cases. The test participant will be presented with a scenario where they have just installed the Kauplur software and now wish to set it up to their liking. The main functionalities of the software are covered by the activities. The common activities include product handling and order processing. There are also some uncommon activities like deleting pages. The participant can skip any step he/she wants. This is necessary because the testers may potentially not have the time needed to complete every step. After each step there is a reference to the use case which this step is used to test. There is also an estimated time that each task should take. This is also done for the customer scenario.

The scenarios which were used while conducting the test can be found in the scenarios subsection of the appendix. They are different only in the sense that they have been translated to Estonian and the use case numbers and approximation information has been removed. All the steps remain the same.

You have just successfully installed your e-shop. The next step as an e-shop owner is to set up your shop according to your own preferences. This is what you set out to do.

- 1. You first choose a language that you prefer to make navigation easier. The shop currently has no name, products or description. The first thing you as a shop owner want to do is change the name of your shop. You think of a name that you feel like fits and change the e-shops name accordingly. (Use case 1, Use case 2) (~3 minutes)
- 2. After changing the name of your shop you wish to give your customers a short description of what your business is about. To do this you want to post the description on your homepage. (Use case 3) (~5 minutes)
- 3. After adding a description to your shop you realize that the default look does not go with the feel of your company. As a result you want to change the background. You change the background to the picture file called “taust.jpg” on your desktop. (Use case 4) (~2 minutes)
- 4. Now there is a new problem, the header and footer background colors do not match your new background image. You pick out a color that you find fit and change them accordingly. (Use case 5) (~7 minutes)
- 5. Finally you are pleased with the appearance of the shop, but there are still no products in the shop. You add 3 new products with a name, regular price, sale price and picture. The picture files can be found on the desktop. (Use case 6) (~15 minutes)
- 6. Now that you have your product in sale perhaps it would be nice to have different categories for your variety of produce. You create 2 categories and put 2 products into the first category and 1 product into the second category. (Use case 7) (~5 minutes)
- 7. You take a look at your new e-shop and find that you do not want the page called “ABOUT US”. You remove it completely. (Use case 8) (~5 minutes)
- 8. Somebody has made a purchase! You go and take a look at the order and change the status to “processing”. (Use case 9) (~ 2 minutes)
- 9. If you feel like you wish to change anything else then please do. If not, then you decide to log out. (~ 1 minute)

Customer scenario

This scenario will cover the customer use cases. The test participant will be conducting this scenario as a customer who has a certain product he/she wishes to purchase.

You are a customer looking to buy a product. Having stumbled upon this e-shop you try to order the product that you are looking to buy. The product you will be trying to buy is an apple.

- 10. Firstly you choose a language that you prefer. You search for the product by name. (Use case 10, Use case 11) (~ 1 minute)
- 11. After finding the product you add it to your shopping cart. You navigate to your shopping cart and realize that you no longer wish to order this item. You remove it. (Use case 12) (~ 4 minutes)
- 12. You navigate back to the shop and choose a different product and add it to your cart. But now you wish to change the amount of the product you wish to order from 1 to 6. You change the amount of product you wish to order. (Use case 13) (~3 minutes)
- 13. Now you are ready to place your order. You successfully place your order. (Use case 14) (~ 2 minutes)

Pre-test questionnaire

The pre-test questionnaire serves as an additional information source about the participant. In the questionnaire the participant is asked about his/her previous experience with the type of software we are testing as well as overall computer skills. It is also used to determine how they are a part of the target audience. This information will be used to add credibility to the data gathered from the participant. Estimated time to complete is 2 minutes. The pre-test questionnaire in Estonian is shown in Figure 1.

Testi-eelne küsitlus

1. Kuidas hindaksid enda arvuti kasutamise oskust?

Väga halvad Väga head

2. Kuidas hindaksid enda teadmisi e-poodide valdkonnas?

Väga nõrk Väga tugev

3. Eelistan vähemate võimalustega ilusama välimusega tarkvara rohkemate võimalustega koledama välimusega tarkvarale.

Ei nõustu absoluutselt Nõustun täiesti

4. Palun kirjeldage kuidas olete seotud kauba müügi või vahetusega?

Done

Figure 1: Pre-test questionnaire

Post-test questionnaire

Sometimes the pre- and post-test questionnaires are the same to see how the test makes a difference in the answers. The post-test questionnaire in this evaluation serves a different purpose from the pre-test questionnaire. It is used to get feedback from the test participant regarding the user experience of the system. The post-test questionnaire is necessary because of the subjective nature of user experience testing. To help quantify emotions and put the observations into context, a reliable scale is needed. It is common to use a 5 or 7 ball scale for measuring which helps in the analysis of the data.

Because of the fact that the test is being recorded there is no need to conduct a very thorough questionnaire where the participant is asked about where and why they got stuck. This can be determined by the evaluator by observing the recordings. Estimated time to complete is 3 minutes. The post-test questionnaire in Estonian is shown in Figure 2.

Testi-järgne küsitlus

1. Kas tarkvara kasutades tekkisid pigem negatiivsed või pigem positiivsed emotsioonid?
 Väga negatiivsed Väga positiivsed

2. Stsenaariumid olid selgelt arusaadavad
 Ei nõustu absoluutselt Nõustun täielikult

3. Tundsin, et stsenaariumid kirjeldasid tegevusi, mida teeksin ka ise e-poe üles seadmiseks.
 Ei nõustu absoluutselt Nõustun täielikult

4. Minu arvates oli süsteem lihtsasti kasutatav
 Ei nõustu absoluutselt Nõustun täielikult

5. Ma tundsin, et ei vaja tarkvara kasutamiseks abi
 Ei nõustu absoluutselt Nõustun täielikult

6. Mul ei tekkinud probleeme ülesannete läbi viimisel
 Ei nõustu absoluutselt Nõustun täielikult

7. Kui tekkisid probleeme ülesannete lahendamiseks, siis mis selle põhjustas?

8. Kuidas hindaksid tarkvara väärtust?
 Väga kalle Väga odav

9. Kas on veel võimalusi, mis meeldiksid, kuid mida hetkel ei ole?

Figure 2: Post-test questionnaire

Test environment setup.

The test environment will be a local host due to the unavailability of a real domain for testing purposes. The local host is the same computer the test will be conducted on. It is the computer's own network service which can be accessed by a web browser. The software will be uploaded along with the database into the local host using a windows web development environment called WampServer [14]. The web browser used during the test will be Mozilla Firefox (version 27.0.1) because it is widely known and used.

In order to set up the test environment, WampServer must first be installed. After successfully installing the software, the next step is to unpack the Kauplur.zip file in the "www" subfolder. The folder can be found in the installation directory of WampServer software. After that the web development environment should be put online. Now it is possible to access the local host by entering the URL "http://localhost" into the web browser. In order to get Kauplur working

properly we have to import the database which is located in a file called wordpress.sql included in the Kauplur.zip file under the wordpress directory. This can be done by going to the local host main page and then clicking the “phpmyadmin” link. Here a new database called “wordpress” should be created. After it is created, importing the wordpress.sql file into the new database shall be the last step. Now by navigating to “http://localhost/wordpress” the test environment can be seen. The administrative page of the e-shop can be accessed from “http://localhost/wordpress/wp-admin” by using the default user and password which both are “admin”. The test environment has been successfully set up.

Tester selection strategy

Testers are a key part of this evaluation. They are needed to gather important information about the user experience. The focus will be on beginner users who are not familiar with the peculiarities of this type of software. When selecting testers it must be taken into account that they should also be a part of the target audience subset. This means testers should be minor Estonian merchants and craftsmen. Each user shall be asked about their previous experience with this type of software. This is done to see if previous experience is a factor in their perception of the system.

According to J. Nielsen’s article “Why you only need to test with 5 users” the correlation curve between usability problems found and number of test users peaks at 5 test users[15]. After the fifth user, observations begin to repeat and discovering new problems becomes scarce. That and a limited time frame is why in this evaluation the test will be conducted on exactly 5 users.

Overview of required resources

During the evaluation there is a need for many resources. The resources and their needs are described in Table 2.

Resource name	Amount	Why it is needed?
Test participant	5	To gather information concerning the usability of the product. We are testing ease of use which we can assess in a small number of participants [4].
Computer	1	A test environment has to be accessed to conduct the test.

Local host uploading software	1	The software is website so it has to be uploaded somewhere. In our case we will be using local host.
Monitoring equipment	1	Needed for monitoring the participants activities.
Evaluation conductor	1	The information has to be gathered and analyzed by somebody.
Web browser	1	Needed for accessing the local network.
Internet connection	1	Needed to connect to the questionnaire website
Questionnaire software	1	For creating and conducting the questionnaire.

Table 2: Resource requirements for the testing

4 Application of evaluation method

In order to evaluate the method that was suggested, it must first be tested. The testing was done on 5 people who are each involved, in different ways, to product marketing. To make the test environment even more genuine, each person was tested in their own home or workplace. This is mostly possible due to the size of Estonia. Going to each test participants home could prove to be difficult in larger countries. This decreases the number of potential test participants as not everybody is willing to let a stranger into their home or workplace. However, because we would be only using five participants, this was not a big issue. Due to the type of test environment, it was impractical to set it up on each test participant's personal computer. All the testing was done on a computer provided by the test conductor. To make sure that the participant will not be interrupted, a time for the testing was agreed on.

To find the right test participants I first researched small businesses and used personal contacts to find people who would fit in to the software's target audience. After finding five people, I could start setting times for the testing. Ideally it would have been best to have all the testing done in one day. Because of the different locations and schedules of the participants, the testing was divided into three days.

Each test session started with the pre-test questionnaire. Completing this usually took 2 minutes, which was expected. After the first survey was filled, I proceeded to explain the think-aloud method and the scenarios to the participant. When the participant was confident on what he/she has to do, the test environment was introduced and the screen and audio recording commenced. When the scenarios were completed, a post-test questionnaire was conducted followed by a small informal talk to get any additional information. After this the recording was stopped and the test was completed. The expected total time for the testing, including the questionnaires, was 1 hour. This estimate based on the number of scenarios and the estimated level of difficulty of the software.

In figure 3 a pre-test state of the e-shop front page can be seen. Figure 4 depicts the expected look of the e-shop after the first scenario has been completed. In the second scenario the look does not change. Figure 5 shows the backend side of the store. That is where all the changes can be made to the looks. Tasks, such as adding and removing products, can also be done from that interface.

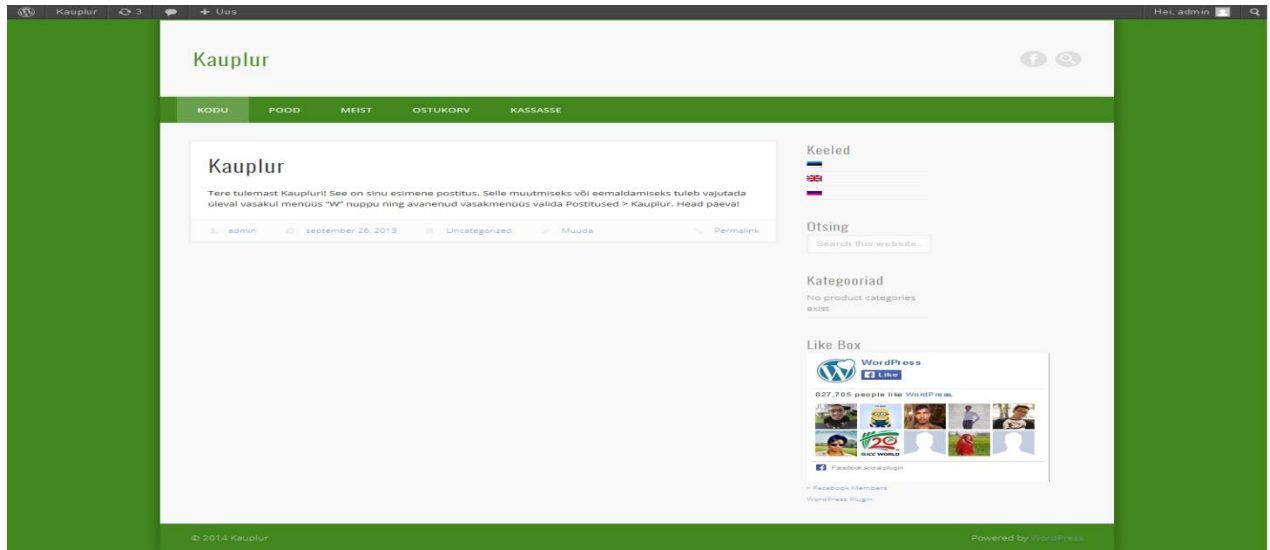


Figure 3: E-shop front page before testing (front end)

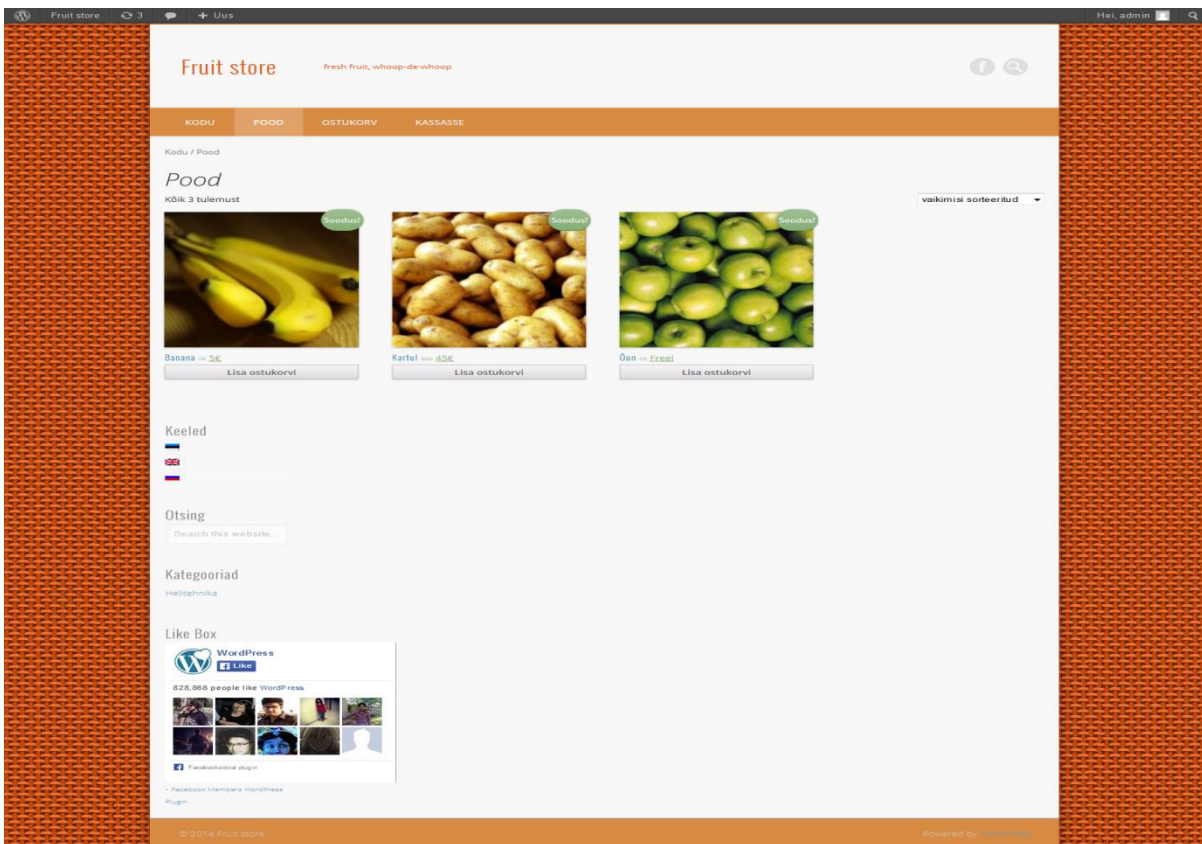


Figure 4: Expected e-shop after first scenario




Kauplur 4 Uus Hei, admin

WordPress 3.9 on saadaval! [Võisid seda kohe uuendada.](#) Braauserisaded Aabina

Tooted Lisa toode

Kõik (3) | Avaldatud (3) | Prügi (9) | Sort Products

Masstegevused Pakenda Kõik kuupäevad Vali kategooria Show all product types Show all sub-types Filter 0 elementi

<input type="checkbox"/>	Name	SKU	Laoseis	Hind	Categories	Tags	★	🛒	Kuupäev	Languages
<input type="checkbox"/>	 Õun	-	Saadaval	4€ 5€	Helitehnika	-	☆	<input type="checkbox"/>	06.04.2014 Avaldatud	Eesti
<input type="checkbox"/>	 Banana	-	Saadaval	7€ 8€	-	-	☆	<input type="checkbox"/>	06.04.2014 Avaldatud	Eesti
<input type="checkbox"/>	 Kartul	-	Saadaval	50€ 45€	-	-	☆	<input type="checkbox"/>	06.04.2014 Avaldatud	Eesti

Masstegevused Pakenda 0 elementi

Atäh, et kasutad WordPressi Hangi versioon 3.9

Figure 5: E-shop back end

5 Results from evaluation

After applying the method on five different people, there is a video and two questionnaires per tester. To evaluate the method, the usability testing should first be analyzed. In order to do this, a usability test results report will be compiled. The report will include the analysis of the data gathered from observations made during the test, questionnaires, speaking with the participants and the recordings.

To identify usability problems, the recordings will all be watched by the test conductor. Since the users are not predictable it is impossible to set up oracles to cover everything. Thus to identify problems the test conductors expert knowledge will be used. During the recordings examination, if something does not match up with what was expected by the conductor, then it will be considered a problem.

The results evaluation could be done differently for different cases. For instance heuristics could be set up prior to the test. Then the scenario steps could be done according to the heuristics under evaluation and then the results compared to the heuristics. Using heuristics would be a more systematic approach and result in more concrete data.

5.1 Results from usability testing

The entire data gotten from the expert analysis of the videos is given in the video observations subsection of the appendix. In this chapter, the most interesting and important issues will be looked over and analyzed. The actual recordings will not be provided with the document due to the large size of the video files. To access them please contact the writer of this paper.

All of the testers were asked to describe their affiliation with small businesses or what they would wish to potentially sell [Table 5 - Q4]. Answers were different – sell handicraft jewels, bakery products, free range chicken eggs and even events and gatherings such as concerts. This made sure, that the testers are indeed potential end users and the sort of people that will be expected to use the product.

Each question that used a scale system, used a five point scale. This makes the average value 3. The average of the scale and the average of the tester's answers can be compared to make observations about the test group. Analyzing each tester individually would take too an extensive

period, so the test group's data is analyzed as a whole, using the average values and reoccurring problems. In Table 3 the questions which use a scale are generalized and the test group average is given. This should give a better understanding of the test group.

Questionnaire data	Test group average
Q1 – Computer skills (1 -Very bad, ..., 5 - Very good)	3.4
Q2 – Knowledge of e-stores (1 – Very little, ..., 5 – Very good)	2.6
Q3 – Prefer looks over functionality in software (1 – Don't agree at all, ..., 5 – Absolutely agree)	3
Q4 – Emotions during the use of the software (1 – Very negative, ..., 5 – Very positive)	3.8
Q5 – Scenarios were easily understandable (1 – Don't agree at all, ..., 5 – Absolutely agree)	4
Q6 – Scenarios described the steps I would do to set up an e-shop (1 – Don't agree at all, ..., 5 – Absolutely agree)	4.4
Q7 – The system was easy to use (1 – Don't agree at all, ..., 5 – Absolutely agree)	3.2
Q8 – I felt like I did not need any help using the system (1 – Don't agree at all, ..., 5 – Absolutely agree)	2.8
Q9 – There were no problems during the tasks (1 – Don't agree at all, ..., 5 – Absolutely agree)	2
Q10 – Rate the looks of the software (1 – Very ugly, ..., 5 – Very good looking)	3.2

Table 3: Questionnaire data

The performance of the testers will be evaluated in terms of success rate, task completion time and number of severe issues reported by the test conductor. To analyze the differences in task completion times, a chart of the task completion time per tester will be given. The selected task was completed by all the testers without skipping or misunderstanding the goal of the task. There were 6 such tasks in total. 3 in the first part and 3 in the second part of the test [Table 7]. In Table 7, the time when each task was completed is given. To calculate the time a task took, the previous tasks completion time has to be subtracted. The time it took to complete task 5 per user is shown in Figure 6. Task 5 involved adding 3 products into the e-shop. This task was expected to take the longest, ~15 minutes.

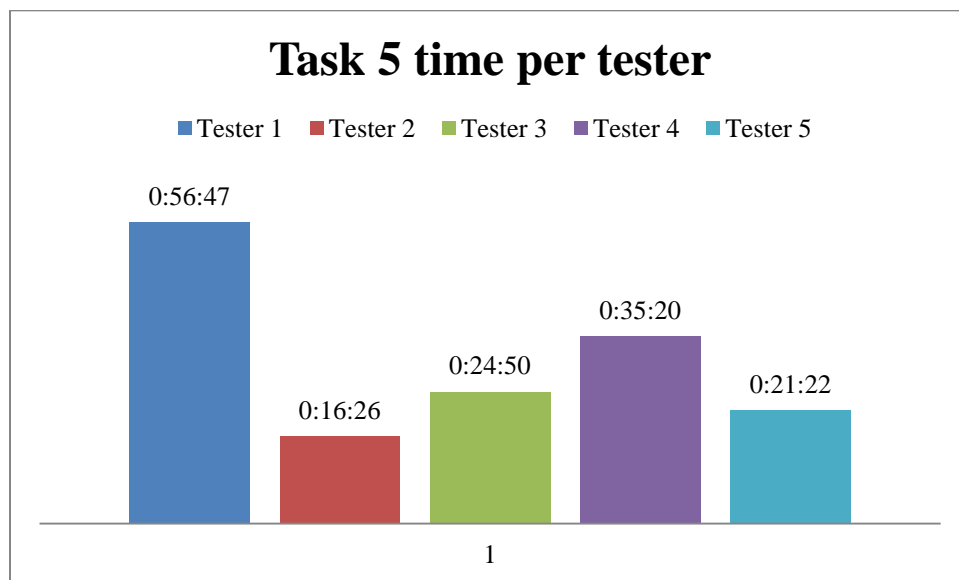


Figure 6: Time to complete task 5 per tester.

The data is rather consistent with the exception of tester 1. This inconsistency is explained by issue #9 under the video observations of tester 1. The slightly higher than expected completion times however indicate that possible usability issues might be present. From looking at the observations we can see, that adding a picture to the products proved to be an occurring problem. This was because the testers could not find the right place where to add the picture to a product. Next the issues that were reoccurring will be brought out and categorized by severity. Reoccurring will be defined as a occurring with 3 or more of the testers. All observed problems can be seen under the video observations section in the appendix.

Because not every problem is the same, a severity scale should be used to determine each problems severity. A severity scale sets each problem into a severity category. The scale can be created by the tester or a previously defined scale could be used. The severity scale that will be used in our case is defined as follows [16]:

- **Critical:** If we do not fix this, users will not be able to complete the scenario.
- **Serious:** Many users will be frustrated if we do not fix this; they may give up.
- **Minor:** Users are annoyed, but this does not keep them from completing the scenario.

In Table 4, the severity of the most occurring problems (occurred in 3 or more of the tests) in our test have been given. This gives a good overview of the software’s overall usability.

Severity	Usability problems
Minor	The tester tries to add products to categories under the categories page.
Serious	The tester does not know which button to use in order to save his/her product to the shop.
Minor	The tester does not know where the correct place is to add a picture for a product

Table 4: Reoccurring problems identified by expert observation

There were few reoccurring problems. This was expected and described in the strategy for selecting testers section. There were no critical errors found.

A problem that was observed the most is that the testers were overwhelmed by the amount of functionalities provided by the software. Testers, in most cases, did not distinguish between the e-shops front end and back end. This confused them and led to misunderstandings in the intent and actual use by the testers.

5.2 Lessons learned about the method

The testers answered that the scenarios were understandable (Table 3 – Q5) and describe a real life situation (Table 3 – Q6). However there were 10 instances where the tester thought that a task was completed when it actually was not or was partially (Table 7). This means that some of the scenario steps should be rewritten so that they are unequivocal.

The expected time for completing the test was 1 hour. Every one of the testers needed more time to complete the test, some even double the time (Figure 7). This is a big problem because the testers might not have enough time to complete the test. From the data gotten from the questionnaire, it can be seen that the average computer skill is 3.4 out of 5. From Table 7 it can be seen that the task which usually took the longest, was adding three products. From the tester notes, also given in the appendix, it can be observed that there is a reoccurring problem during the time when the testers are adding products. The problem is described in several occasions as follows – “The tester gets immersed in the products page and starts adding additional info not requested in the scenarios. This increases the total time of the test” [Appendix Video observations – Tester 1 problem #11, Tester 2 problem #7, Tester 3 problem #9]. In this case, the issue could have been avoided by explaining to the tester that only the tasks that are described should be done and nothing else. That would have avoided the testers from exploring features that were not a part of the test.

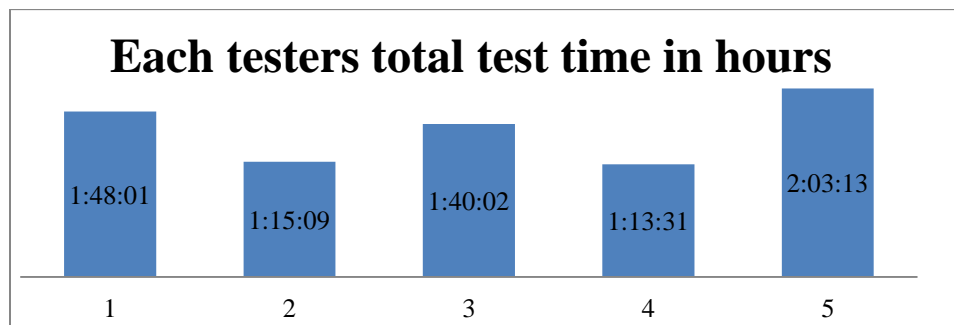


Figure 7: Test times

Another problem was that the testers were not able to complete the tasks in reasonable time. The maximum amount of time that any of the tasks should have taken was 30 minutes. In some cases this held up, but in some of the tests the person would be stuck on one step for over an hour. In those cases the conductor should involve and ask the tester to either move on to the next step or give the tester enough information so that the task can be completed. During the testing done for this thesis, the conductor had to help two of testers. In both of the cases, the testers made an assumption about the system and were very patient. To help prevent this problem, the testers should be made aware of any bugs or particularities of the software. Also the test conductor should give hints when any task takes longer than expected. This means that each of the tasks

should have an expected time of completion. This could also be useful for individual task analysis.

Conclusion

As a result of the questionnaire and think-aloud method with screen recording, there is a huge amount of various data that is captured using this method. Due to the variety of data (audio, activities etc.) it is easy to compare the users intended use and the actual use of a system, or any other sort of comparison for that matter. The scenario steps make the intended use more systematic. In this case however, the scenario steps were not unequivocal enough. There was also a static noise on each of the recordings. A pilot tester should be used to avoid this type of problems.

This method is effective at finding the usability problems in the tasks that are presented. They do not however give a complete overview of the entire systems usability. This could be achieved by including every possible task in the scenarios, but that might be impractical for e-shops that have a lot of functionalities. For the case presented in this thesis, setting up scenarios proved to be good – it set a rough time frame for each test by limiting the users' curiosity. The use of 5 test users makes this method require few resources. For small software firms or individual software developers this method might provide accurate data for a comprehensive evaluation of their e-shop software for a reasonable price.

6 Summary

The use of e-shops has been growing exponentially. This has led to people wishing to sell their own handicraft or other sorts of products in a personal e-shop or store. This has led to many different pieces of software that allow the user to set up their shop. For a novice user this might prove to be hard. This is why the software makers should make sure that their products are easily usable.

The objective of this bachelor's thesis is to suggest a method, which the software makers could use to make their software more learnable, memorable, efficient, satisfying and error free.

During the development of this thesis, I have conducted a research of different methods of usability testing to derive a method that could potentially bring a lot of good data for later analysis. This was done while bearing in mind that the method will also be conducted.

The suggested method was used on 5 test users and an e-shop software called Kauplur, which was co-developed by the author of this thesis. This resulted in 8 hours of video and audio recordings. The description of the test application is given for a better understanding. The recordings were then analyzed and results about the usability of Kauplur were given.

The last part of this thesis looks at the lessons learned from conducting the test using the suggested method for evaluating e-shop software.

Although the method led to the discovery of numerous unexpected usability problems, it is limited to the scenario tasks. For a more complete overview the testers should not be restricted when setting up the store.

7 Kokkuvõte

E-poodide kasutus on viimaste aastatega eksponentsiaalselt kasvanud. Sellest tingituna on tekkinud rohkem indiviide, kes sooviksid oma käsitööd või muud sorti tooteid personaalses e-poes. See on viinud paljude erinevate tarkvaradeni, mis lubavad kasutajal üles seadistada oma enda e-poe. Algajakasutajale võib see olla liiga raske, millest tulenevalt jäetakse e-poe tegemine pooleli. Selleks, et nende toodet kasutataks, peaksid tarkvaraarendajad kindlaks tegema, et toode on lihtsasti kasutatav.

Selle bakalaureusetöö eesmärgiks on pakkuda välja meetod, mida tarkvaraarendajad saaksid kasutaja selleks, et oma toodet muuta kergemini õpitavaks, meeldejäavamaks, efektiivsemaks, rahuldust pakkuvamaks ja vigade vabamaks.

Bakalaureusetöö käigus uuriti erinevaid meetodeid kasutatavuse testimiseks. Seda selleks, et nendest meetodidest välja tuletada üks meetod, mis tooks arendajatele kasulikke andmeid, et nad saaksid eelpool mainitud omadusi parandada. Meetodi tuletamisel peetakse meeles ka seda, et hiljem kavatakse seda meetodid päris testimisel rakendada.

Tuletatud meetodit kasutati 5-1 potentsiaalsel lõppkasutajal, kasutades tarkvara nimega Kauplur. Selle tarkvara kaasarendas selle töö autor. Testimisel tulemusena saadi 8 tundi video- ja helilindisusi. Meetodi paremaks arusaamiseks on antud testimise rakendamise kirjeldus. Seejärel analüüsiti lindistusi ning Kaupluri kasutatavuse tulemused pandi kirja.

Töö viimases osas vaadatakse üle meetodi kohta õpitud õppetunnid ning pakutakse mõningaid viise kuidas testimistehnikat parandada.

Kuigi seda meetodit kasutades leiti mõningaid ootamatuid kasutatavuse probleeme, siis olid need stsenaariumite punktidega piiratud. Täielikuma ülevaate saavutamiseks, ei tohiks testijatele poe sätestamisel piire panna.

References

- [1] <http://www.epa.gov/evaluate/glossary/c-esd.htm> [Accessed: 12.04.2014]
- [2] <http://www.nngroup.com/articles/usability-101-introduction-to-usability/> [Accessed: 12.04.2014]
- [3] ISO FDIS 9241-210:2009. Ergonomics of human system interaction - Part 210: Human-centered design for interactive systems (formerly known as 13407). International Organization for Standardization (ISO)
- [4] Mahmut Ekşioğlu, Esin Kırış, Tuğba Çakır, Merve Güvendik, Efsane D. Koyutürk, and Merve Yılmaz. 2013. A user experience study of airline websites. In *Proceedings of the Second international conference on Design, User Experience, and Usability: web, mobile, and product design - Volume Part IV* (DUXU'13), Aaron Marcus (Ed.), Vol. Part IV. Springer-Verlag, Berlin, Heidelberg, 173-182. DOI=10.1007/978-3-642-39253-5_19 http://dx.doi.org/10.1007/978-3-642-39253-5_19 [Accessed: 28.04.2014]
- [5] <http://wordpress.org/themes/pinboard> [Accessed: 20.04.2014]
- [6] J. Nielsen and R. Molich,(1990). Heuristic evaluation of user interfaces. Available from:<http://dl.acm.org/citation.cfm?id=97243.97281&coll=DL&dl=GUIDE&CFID=304122846&CFTOKEN=86157456> [Accessed: 12.04.2014]
- [7] J. Nielsen and R. Molich,(1990). Improving a human-computer dialogue. Available from:<http://dl.acm.org/citation.cfm?id=77481.77486&coll=DL&dl=GUIDE&CFID=304122846&CFTOKEN=86157456> [12.04.2014]
- [8] J. Nielsen (1994) Usability Inspection Methods.
- [9] J. Nielsen (1994). Usability Engineering.
- [10] Law, E., Roto, V., Hassenzahl, M., Vermeeren, A., Kort, J. Understanding, Scoping and Defining User Experience: A Survey Approach, 2009 Available from: <http://dl.acm.org/citation.cfm?id=1518701.1518813&coll=DL&dl=GUIDE&CFID=422273901&CFTOKEN=26260488> [Accessed: 12.04.2014]

[11] bitbucket.org/tarkvaraprojekt/kauplur/wiki/Funktsionaalsed%20n%C3%B5uded
[Accessed: 12.04.2014]

[12] <http://www.kaner.com/pdfs/ScenarioIntroVer4.pdf> [Accessed: 24.04.2014]

[14] www.survermonkey.com [Accessed: 12.04.2014]

[13] www.techsmith.com/camtasia.html?gclid=CP7Co531qL0CFQkUwwodRLIAMAð
[Accessed: 12.04.2014]

[14] <http://www.wampserver.com/en/> [Accessed: 28.04.2014]

[15] <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
[Accessed: 12.04.2014]

[16] [http://www.usability.gov/how-to-and-tools/methods/reporting-usability-test-
results.html](http://www.usability.gov/how-to-and-tools/methods/reporting-usability-test-results.html) [Accessed: 28.04.2014]

Appendix

I. Test result analysis

This appendix includes the data gathered from analyzing the 5 tester's videos. The videos consisted of the tasks done during the scenarios where the computer screen was recorded along with the audio captured from the testers use of the think-aloud method. The data is in the original language which is Estonian.

Question	Person 1	Person 2	Person 3	Person 4	Person 5	AVG
Q1	3	3	4	4	3	3,4
Q2	3	3	3	3	1	2,6
Q3	3	4	3	4	1	3
Q4	Toodan põllumajandustooteid ja vahendan infot teiste tootjate toodete kohta potentsiaalsetele tarbijatele.	Olen seni olnud peamiselt kauba ostmisega. Ise ei ole müünud.	Oman ettevõtet, mis toodab ja müüb nii kaupa kui ka teenust	Käsitöötarb eid oleks tore netis müüa	Helitehnika rentimine on mu igapäeva töö	
Q5	4	1	4	5	5	3,8
Q6	5	4	4	4	3	4
Q7	5	4	4	4	5	4,4
Q8	2	1	4	5	4	3,2
Q9	2	2	2	5	3	2,8
Q10	1	1	3	2	3	2
Q11	osaliselt enda süvenemine, st lugemisoskus ja arusaamine. teisena see, et osade käskude ja nuppude kasutamine oli raske, kuna keel oli tundmatu ja ei suutnud mõista, mis nupu all tegelikult peidus	Ei osanud piisavalt palju ise uurida, kuidas salvestada, muutusi teha.	Tahtsin endas kindel olla ja küsisin abi	Arusaamat us valedetele asjadele vajutades	-	
Q12	2	3	3	4	4	3,2
Q13	lihtsalt see et pildi kõrval võiks olla toote kirjeldus, mitte all ja et ei peaks kerima	Tegevuste jaotused võiksid kõik olla ühel pool, vasakul. Paremalt olevad nupud ja tegevused jäid pidevalt märkamatuks.	Pildifailid võiks olemas olla - taust näiteks. See annab kiiresti pood valmis teha ja seejärel tuttavatel testida. Vajadusel saab seejärel töötavat e-poe välimust muuta	Ei oska öelda	-	

Table 5: Questionnaire results

Pre-test questionnaire questions

Post-test questionnaire questions

The corresponding question (in Estonian) to each mark in the first column:

- Q1 – Kuidas hindaksid enda arvuti kasutamise oskust?
- Q2 – Kuidas hindaksid enda teadmisi e-poodide valdkonnas?
- Q3 – Eelistan vähemate võimalustega ilusama välimusega tarkvara rohkemate võimalustega koledama välimusega tarkvarale.
- Q4 – Palun kirjeldage kuidas olete seotud kauba müügi või vahetusega?
- Q5 – Kas tarkvara kasutades tekkisid pigem negatiivsed või pigem positiivsed emotsioonid?
- Q6 – Stsenaariumid olid selgelt arusaadavad
- Q7 – Tudsin, et stsenaariumid kirjeldasid tegevusi, mida teeksin ka ise e-poe üles seadmiseks
- Q8 – Minu arvates oli süsteem lihtsasti kasutatav
- Q9 – Ma tundsin, et ei vaja tarkvara kasutamiseks abi
- Q10 – Mul ei tekkinud probleeme ülesannete läbi viimisel
- Q11 – Kui tekkis probleeme ülesannete lahendamise, siis mis selle põhjustas?
- Q12 – Kuidas hindaksid tarkvara välimust?
- Q13 – Kas on veel võimalusi, mis meeldiksid, kuid mida hetkel ei ole?

Time to complete test per tester	
1:48:01	Tester 1
1:15:09	Tester 2
1:40:02	Tester 3
1:13:31	Tester 4
2:03:13	Tester 5
1:35:59	Average
7:59:56	Sum

Table 6: Total test time

Time when each tester completed each task						
Scenario task number	Tester 1	Tester 2	Tester 3	Tester 4	Tester 5	
1	0:01:57	0:09:30	0:19:10	0:14:40	1:00:04	First part
2	0:02:27	0:18:50	0:26:45	0:10:17	1:01:13	
3	0:11:45	0:20:40	0:28:58	0:12:01	0:59:47	
4	0:18:45	0:23:31	0:38:01	0:15:52	1:02:25	
5	1:11:11	0:45:14	1:02:51	0:51:12	1:23:47	
6	1:17:42	0:54:59	1:18:44	0:55:33	1:35:01	
7	1:18:19	1:03:26	1:20:01	0:59:00	1:41:50	
8	1:28:13	1:03:26	1:26:45	1:00:03	1:45:38	
9	1:29:43	1:03:50	1:26:59	1:04:04	1:45:40	
10	1:31:07	1:04:00	1:27:04	1:05:01	1:51:46	Second part
11	1:37:23	1:04:27	1:30:42	1:05:29	1:53:15	
12	1:37:53	1:05:27	1:31:53	1:05:58	1:54:03	
13	1:44:30	1:06:43	1:34:21	1:06:20	1:55:40	
* - Skipped [by accident(forgot) or intentionally]						
** - Tester thought that the task had been completed when it actually was not or was partially						

Table 7: Tasks completion times per tester

II. Video observations

In this section, notes made during each video are given in a table format where the time when the problem first occurred is under the time column, the expected behavior and the actual behavior are described under corresponding columns. Finally, the problem description is given, where I tried to write down why this sort of behavior should be considered a problem, as well as additional notes about the issue.

1. Tester 1

Video 1 - total time 1:48:01				
#	Time	Expected behavior	Actual behavior	Problem description
1	0:00:00	Crisp audio	Audio is noise polluted	High noise from audio recording due to internal microphone and noisy fan
2	0:03:56	The tester takes the background image from the desktop and changes the stores background	The tester begins browsing all the pictures on the desktop and tries to set the background.jpg image as the desktop background not e-shop background	The tester does not understand the scenario description possibly because of specialty text (e.g. Desktop). Also it was not clear to the tester that the actions have to be done through the e-shop interface.
3	0:06:40	Tester uses the web-browser to carry out the scenarios	The tester can't find the web-browser	The tester could not find the web-browser due to the peculiarity of the test conductor's computer. This forced the conductor to interfere with the test, but should not be considered an issue. This was also possibly caused by the need to click between the "scenarios" word document and the web browser with the test environment. Could be prevented if the scenarios were presented on paper carrier.
4	0:06:57	The tester changes the store name and description	The tester forgets about the first and second points in the scenario description and moves on to changing the background	The tester skips a step even though he/she thinks of the name and a description for the shop as heard from the think-aloud method. This is due to the previous problem (#3).
5	0:09:51	To change the background the tester moves to the back end of the software.	The tester tries to change the background by clicking the "change" button for the first page post.	The tester found the first text called "change" and tries to click on it. Realizes straight away from the tooltip that this will change the post and not the background. Could be avoided by removing the clutter text from posts.

6	0:10:07	The tester realizes he/she is in the backend of the shop.	The tester accidentally navigates to the backend and instantly clicks the "back" button on the web browser.	The tester does not realize that he/she is in the right place and navigates back to the front page.
7	0:14:15	The tester changes the header and footer color	The tester changes the background color and header text color	The tester thinks that the background color represents the header and footer color, but in reality they are separate. Moving the header and footer color selection into the customize page could have prevented this problem.
8	0:29:44	To add a product to your shop page the "avalda" button is pressed	The "salvesta mustandina" and afterwards "eelvaade" button are pressed to add a new product	The tester does not know which button to use in order to save his/her product to the shop.
9	0:30:11			There is an error with the program that shows the "save" gif even when not saving anything. The tester thinks that their work is being saved and waits. Test conductor involves (at 00: in order for the test to go on, because the tester does not realize that there is a problem and waits patiently.
10	0:49:45	Adding a picture for the product is done from the right hand side menu under the segment "Tunnuspilt"	The picture is added into the description.	The tester knows where the picture should be inserted in the products page, but does not know where to add in under the product back end page.
11	0:53:21	The tester only uses the shop to play out the steps described in the scenarios	The tester does tasks not described in the scenarios	The tester gets immersed in the products page and starts adding additional info not requested in the scenarios. This increases the total time of the test.
12	1:07:20	The pages are fully translated into the language selected	The pages are not fully translated to the language selected	The "categories" page is only partially translated into Estonian and this interferes with the tester while he/she is trying to add new categories.
13	1:11:49	Products are added into categories under each products page	The tester tries to add products to each category under each categories page	Tester tries to add products to categories under the categories page.

14	1:15:25	The shop owner knows that after pressing "update" a product is updated with the new settings.	The tester does not realize that after pressing the "update" button the product has been added to a category.	After adding a categories to the product, the "uuenda" button is pushed and the product is now added to the category (is) selected. A visual queue should be shown to indicate that this action has been completed.
15	1:20:52	To find the orders the shop owner goes to WooCommerce > Orders	The tester can't find the orders in the left hand side menu	The tester can't find the orders page because he/she does not know to look under the WooCommerce menu. Even when that is done, the page is not translated and the person can't find the orders page.
	1:30:00	First part completed		
16	1:31:47	After adding a product to the shopping cart, the customer can continue shopping in the shop	The tester expects to be forwarded to the shopping cart page.	The tester expects to be forwarded to the shopping cart page.
17	1:36:02	The product name is visible to the customer	The customer can't find a product because he/she is unable to see the product name in the shop page	The product name under the products picture is too small
	1:44:30	Second part completed		

2. Tester 2

Video 2 - total time 1:15:09				
#	Time	Expected behavior	Actual behavior	Problem description
1	0:00:00	Crisp audio	Audio is noise polluted	High noise from audio recording due to internal microphone and noisy fan
2	0:06:13	To change to store name one must go to the customize page	The tester tries to change the store name by changing the first page post title. However adding the description is done.	The tester goes straight to the second step in the scenarios while thinking that they are doing the first step. 3
3	0:13:46	To save changes the "Uuenda" button has to be clicked	The testers leaves the page thinking that the changes that were made were saved automatically	The tester does not realize that in order to save the changes that were made, the "uuenda" button has to be pressed. The changes are not saved automatically.

4	0:22:40	The tester changes the title and short description of the store	The tester does not realize that they are in the customize page and doesn't change the title.	
5	0:26:31	To save products the "Avalda" button has to be clicked	The tester clicks the "Salvesta mustandina" button	To save a new product to the store the tester click the button called "Salvesta mustandina" instead of "Avalda". Could be because "salvesta mustandina" contains the word "Salvesta" which means "save" and in the think-aloud method the tester is saying "I wish to save the product now"
6	0:27:01	Adding a picture for the product is done from the right hand side menu under the segment "Tunnuspilt"	The picture is added into the description.	The tester knows where the picture should be inserted in the products page, but does not know where to add in under the product back end page.
7	0:28:11	The tester only uses the shop to play out the steps described in the scenarios	The tester does tasks not described in the scenarios like adding the stock state.	The tester gets immersed in the products page and starts adding additional info not requested in the scenarios. This increases the total time of the test.
8	0:32:29	The tester uses the "lisa toode" button to add each product	The tester creates one product and then duplicates it and changes the duplicates information	The tester found a fast way for adding products by duplicating the first product they added. This is not a problem, but rather a remark.
9	0:42:18	Products are added into categories under each products page	The tester tries to add products to each category under each categories page	Tester tries to add products to categories under the categories page.
10	0:48:35	The tester follows the scenarios exactly	The tester is diverted from his/her task by the numerous possibilities provided by the platform	The tester adds additional fields to the product which are unnecessary. This also increases the time of the test.
11	0:53:35	To update a product all the changes are made and then the "Uuenda" button is pressed	After each change the "uuenda" button is pressed.	To ensure that all changes are saved the tester click the "uuenda" button after every change. Not realizing that you can make many changes and only after you are happy with all of them should you click the button. This would save time.
	1:01:15	First part completed		
12	1:04:52	The tester uses the search bar to find a product	The tester goes to store and does not use the search option.	The scenario step is skipped and it is impossible to tell if the tester would have encountered problems.
	1:10:43	Second part completed		

3. Tester 3

Video 3 - total time 1:40:02				
#	Time	Expected behavior	Actual behavior	Problem description
1	0:00:00	Crisp audio	Audio is noise polluted	High noise from audio recording due to internal microphone and noisy fan. Reoccurring issue in all the tests. Won't be noted any more.
2	0:05:40	To go to the store backend the gray menu on the very top of the website is used	Pressed on "admin" button under post. Found the correct button at 0:06:14	"W" button on the left side corner is hard to find.
3	0:13:03			Navigating through unrelated places
4	0:14:33	Clicking the language in the left side menu changes the language	Tester clicks on the languages but does not realize that the language has been changed	After the language selection, the tester still tries to find a place to change the language. This is due to the partial translation of the software.
5	0:16:41			Tester accidentally (by pressing a key binding) opens a new tab with unrelated content. Does not close it. This could have potentially disturbed the tester.
6	0:23:13	To save a post the "uuenda" button has to be pressed	The tester has trouble finding the "uuenda" button and leaves the page without saving	The tester does not know where to save the changes that were made. Navigating away from the page gives an error that you're data will not be saved but the tester leaves anyways and later realizes that they have to redo the things they did before.
7	0:31:09	The tester changes the header and footer color	The tester changes the background color and header text color	The tester thinks that the background color represents the header and footer color, but in reality they are separate. Moving the header and footer color selection into the customize page could have prevented this problem.
8	0:32:20	To add a product the "Toode" menu is used	Tester tries to add a product by adding a new post.	The tester thinks that adding a new post is the same as adding a new product. The tester also says that he/she expected that for some reason.
9	0:45:15	The tester only uses the shop to play out the steps described in the scenarios	The tester does tasks not described in the scenarios	The tester gets immersed in the products page and starts adding additional info not requested in the scenarios. This increases the total time of the test.
10	0:46:28	Adding a picture for the product is done from the right hand side menu under the segment "Tunnuspilt"	The picture is added into the description.	The tester tries to add a picture from the "Product Gallery" column.

11	0:50:37	To upload a picture the "Laadin üles" link is used or the pictures are dragged and dropped into the "meediateek"	The tester is in the "meediateek" but does not realize that they have to upload the files from the desktop	The tester can't find the link on which to click to add the pictures into the "meediateek". Uses pictures already present and adds them as the product picture.
12	1:22:01	To delete the "Meist" page it should be removed from the header menu	The tester believes that deleting the page under "lehed" has deleted the menu item	The scenario is not clear on what exactly should be removed. There should be a specification that the Menu item called "Meist" has to be removed and not the page's insides.
13	1:26:03	To change any order to "töötlemisel" state, it has to be first selected	The tester does not select the order and tries to set the state to the one asked	The tester doesn't understand the table concept and doesn't read the error message given after he/she does an action wrongly.
	1:26:45	First part completed		
14	1:29:44	The tester uses the search bar to find a product	The tester goes to store and does not use the search option.	The scenario step is skipped and it is impossible to tell if the tester would have encountered problems.
		Second part completed		

4. Tester 4

Video 4 - total time 1:13:31				
#	Time	Expected behavior	Actual behavior	Problem description
1	0:05:00	To go to the store backend the gray menu on the very top of the website is used	Pressed on "admin" button under post.	The tester's do not read the post which contains the instructions to how to change the post and how to get to the backend of the store.
2	0:07:02	The tester only adds the Estonian title for the post	The tester adds both English and Russian translations	The tester spends time adding unnecessary headings and searching for translations on google translate. This lengthens the test period.
3	0:12:51	The tester changes the header and footer color	The tester changes the background color and header text color	The tester thinks that the background color represents the header and footer color, but in reality they are separate. Moving the header and footer color selection into the customize page could have prevented this problem.
4	0:17:51	The tester follows the scenarios exactly	The tester is diverted from his/her task by the numerous possibilities provided by the platform	The tester accidentally installs a new theme which is not requested in the scenarios. Later he/she changes it back at 0:27:30. This is an advanced feature provided by the WordPress platform which could potentially alter the entire store's interior. Theme options should be hidden.

5	0:30:57	Adding a picture for the product is done from the right hand side menu under the segment "Tunnuspilt"	The picture is added into the description.	The pictures are added into the "meediateek" and then to the description box of the product instead of the "tunnuspilt" section where it is supposed to go.
6	0:31:56	See #4	See #4	The tester adds a date period for the "on-sale" price.
7	0:33:18			Tester has changed the front page from "Kodu" to "Meist". Realizes this and goes to customize the store and changes the front page back to "Kodu" at 0:34:16. This indicates that the tester has inquired understanding of the system.
8	0:53:09	Products are added into categories under each products page	The tester tries to add products to each category under each categories page	Tester tries to add products to categories under the categories page.
9	0:57:18	Tester doesn't check if his/her actions in the backend have had effect in the frontend	Tester checks that what he/she wished to do, worked.	Tester checks if the actions he/she did in the backend have been effective in the store side. If it has not, he/she tries to do it the way that was understood from the scenario. This is also the correct way.
	1:04:04	First part completed		
10	1:03:00	Scenarios dictate that after completing the first part the user should log out	The users log out	In order to complete a purchase, a user has to be logged in and this interferes with the tests second part.
	1:06:23	Second part completed		

5. Tester 5

Video 5 - total time 2:03:13				
#	Time	Expected behavior	Actual behavior	Problem description
1	0:12:14			The tester tries to change the post by clicking on it, hoping it would possibly open up an edit-mode
2	0:13:01	The tester saves his/her changes	Leaves the customize page without saving the changed title, description and new background	The scenario steps are not completed if the changes that the tester does are not saved.
3	0:42:50	The tester follows the scenarios exactly	The tester is diverted from his/her task by the numerous possibilities provided by the platform	The tester adds a comment to the first page post

4	0:44:39	The user does not log out during the test	The tester accidentally logs out	The user is not familiar with the username and password of the system and thus is unable to log back in after logging out.
5	0:52:33	The tester saves his/her changes	Leaves the posts page without saving the changes	The scenario steps are not completed if the changes that the tester does are not saved.
6	0:58:01	The test conductor does not interfere with the tester	Conductor shows the tester, that he/she has to save the changes made.	After 58 minutes without any changes, the test conductor interferes and gives a hint to the tester on how to go on with the test - "changes have to be saved, they are not automatically saved". Also shows that the changes are made from the back of the store.
7	1:18:00	Adding a picture for the product is done from the right hand side menu under the segment "Tunnuspilt"	The picture is added into the description.	The tester tries to add a picture from the "Product Gallery" column.
8	1:18:25	To add a product to your shop page the "avalda" button is pressed	The "salvesta mustandina" and afterwards "eelvaade" button are pressed to add a new product	The tester does not know which button to use in order to save his/her product to the shop.
9	1:31:15	Products are added into categories under each products page	The tester tries to add products to each category under each categories page	Tester tries to add products to categories under the categories page.
10	1:33:58	"Uuenda" button is only pressed after any changes are made	Tester presses the "uuenda" button for no reason	The tester hasn't understood the functions of the buttons.
11	1:35:01			Created categories, but didn't add the products into the categories
	1:45:48	First part completed		
12	1:53:46	After adding an item to the shopping cart, the tester is familiar with how to add items to the shopping cart.	Tester scrolls down the shop page to try and find a way to add items to the shopping cart even after having added some before.	The tasks are not easily redo able or learnable
13	1:55:30	Tester picks the payment method they want	Tester picks a random payment method	The payment methods are not understood by the testers. One reason might be because they have not been translated. Another might be insufficient tooltip information.
	1:55:45	Second part completed		

III. Additional notes

Tester 1	Tester 2	Tester 3
<p>The tester does not read the scenario descriptions thoroughly. This causes him/her to accidentally skip steps. She/he has trouble navigating in the right hand side bar and in the computer's operating system as a whole. This could be caused by a low computer skill. Good use of the think-aloud method. Expressed thoughts loud and clear</p>	<p>The tester speaks in a low voice and due to the internal microphone and noise; it is difficult to interpret the speech in later analysis. The think-aloud method is not used as much compared to other testers.</p>	<p>After finding the backend of the store, the tester tries to briefly adjust to it by reading through the menu. Understand the think-aloud method very well.</p>
Tester 4	Tester 5	
<p>Tester installs a new theme but after 10 minutes of browsing, is able to switch back to the original theme and continues to do the scenario assignments. Hard to hear because of low voice. Doesn't have a problem with finding a way to save things. Prefers trying everything to reading and trying to understand. At 0:50:30, says that adding a picture to the product is difficult. Tester says at the end of the test that playing out the scenarios was enjoyable and interesting.</p>	<p>Gets lost for the first 50 minutes of the test, having not completed any of the tasks given. Mainly because of not saving the changes that he/she made. Doesn't realize that changes have to be saved and keeps making the same mistake until conductor has to interfere. Also has trouble with English and doesn't find a way to change the language - this was the very first step. In the second part, has a good understanding of the scenarios.</p>	

IV. Scenarios

E-poe omanik

Sa oled just installeerinud oma e-poe. Järgmiseks sammuks on enda eelistuste järgi e-poe üles sättimine. Seda sa hakkadki tegema.

- Esiteks valid sa endale meelepärase keele selleks, et lihtsalt ja arusaadavalt navigeerida. Hetkel on sinu poe nimeks Kauplur ning seal puuduvad tooted ja poe kirjeldus. Poe omanikuna soovid sa esimese asjana muuta oma e-poe nime. Sa mõtled välja sobiva nime ning vahetad e-poe nime selleks.
- Peale nime vahetust soovid sa lisada oma klientide jaoks lühikese kirjelduse oma ärist ning millega sa tegeled. Selleks muudad sa oma esilehel oleva postituse enda poe kirjelduseks.
- Kirjelduse lisamisele järgnevalt leiad sa, et poe välimus ei ole see, mis sinu arvates sobib. Selleks muudad sa ära poe tausta. Sul on töölaual olemas pilt nimega “taust.jpg”, mille paned nüüd enda poe taustaks nii, et see katab ühtlaselt terve tagatausta.
- Nüüd tekkis uus probleem, päise ja jaluse värv ei sobi uue taustaga. Selleks vahetad nende värvid ära selliseks, mis sinu arust sobivad uue tausta pildiga.
- Lõpuks oled sa oma poe välimusega rahul, kuid hetkel puuduvad poest tooted. Sa lisad 3 uut toodet pannes neile sinu arust sobivad nimed, tavahinna, soodushinna ning pildi. Pildid oled sa juba välja valinud ning töölauale paigutanud nimedega “oun”, “banaan” ja “kartul”.
- Nüüd kui sul on poes tooted müügil, soovid sa neid kategooriatesse jagada. Selleks lood sa 2 uut kategooriat sinu arust sobivate nimedega ning lisad esimesse kategooriasse 2 toodet ning teise kategooriasse 1 toote.
- Oma e-poele silma peale visates leiad sa, et üks lehtedest tundub sulle ülearu. Sa eemaldad lehe “MEIST”.
- Keegi on teinud ostu! Sa lähed vaatad tellimusi ning muudad oma ainsa tellimuse oleku “töötlemisel”.
- Kui jäi veel asju, mida sooviksid teha, siis võid neid teha. Kui tunned, et kõik sai nii nagu vaja, siis logi välja.

Klient

Sa oled klient, kes otsib ühte kindlat toodet, mida ta soovib osta. Satud siia e-poodi ning proovid seda toodet osta. Toode, mida sa proovid osta on õun.

- Esiteks valid sa endale sobiva keele. Seejärel proovid toodet otsinguriba kasutades leida.
- Pärast toote leidmist lisad sa selle enda ostukorvi. Seejärel otsustad, et sa ikkagi ei soovi seda toodet ning eemaldad selle enda ostukorvist.
- Järgmiseks navigeerid sa poe lehele ning valid õuna asemel kartulit. Sa lisad selle enda ostukorvi. Otsustad ka sõpradele kartulit osta ning muudad toote hulka.
- Nüüd oled valmis tellimuse esitamiseks. Saadad edukalt tellimuse ära.

V. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Erik Berendsen** (date of birth: 07.04.1992),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
 - 1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Usability testing of e-shop software,

supervised by Dietmar Alfred Paul Kurt Pfahl,

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **14.05.2014**