Set the User Free

Exploring the difference between exploratory and taskbased remote asynchronous usability testing

Stine Eklund

Interaktive Digitale Medier

Aalborg University

October 2016

AALBORG UNIVERSITY

Stine Eklund

Set the User Free

Master Thesis

Abstract

The purpose of this paper was to make a comparative analysis of two different walkthrough methods for Remote Asynchronous Usability Testing; taskbased walkthrough method and an exploratory walkthrough method. The paper sought to examine differences, not only in terms effectiveness, but also in terms of subjective opinion towards the website being tested. Likewise the paper sought to explore any differences in testers ability to make use of the exploratory walkthough method and if found, if it is possible to categorize these.

Among the key findings the study found no significant ($p = 0.46$) difference in the subjective opinion of the two groups towards the website. The study also found that, while there was no significant ($p = 0.786$) difference in the amount of errors each method identified, they discovered errors at different parts of the shopping process. The exploratory method identified all of its unique errors during the first part; browsing and shopping, while the task method identified most of its unique errors during the last part; buying and check-out. This is assumed to be due to the task method group being forced through to the check-out process, while many in the exploratory method group didn't. Suggestions on how to improve upon this are covered in the Discussion.

Acknowledgements

I would like to thank my supervisor Tem Frank Andersen for his input and patience through this project. I'd also like to thank my family and boyfriend for their love and support. Lastly I want to thank UserTribe for providing testers during the project and for a lovely intership. Thank you!

Contents

Chapter 1

Motivation

The idea for this thesis came during an internship at the Copenhagen based company UserTribe. UserTribe conducts usability testing of websites and applications. The main product of the company is the remote asynhronous usability test. These tests are performed by the tester at home without any supervision and while using the thinking-aloud protocol.

By using a screenrecorder and a microphone the users record their screen and their voice. This is done while solving 5-7 tasks, lasting 20-30 minutes in totel. The tasks are usually designed to lead the tester through relevant areas of the website. E-commerce websites will for example have tasks that promp the tester to browse for an item, add the item to the basket and lastly go through the checkout flow. Websites that provide information such as the website of a municipality will have assigments with a focus on finding information, both by using the search field and navigating through menus. When the testers are done recording, an evaluator goes through the videos, notes down any errors encountered and selects videoclips that shows users encountering an error and, most importantly, his reaction to it.

1.1 Pitch Test

During my intership i was tasked to set up a number of so-called pitch tests. A pitch test consist of a single tester going through a potential clients website. The video produced is then used at the preliminary sales meeting with the client to show off the method and to make it relatable.

However at one point, there was feedback stating that the pitch test videos seemed fake which rendered them useless. The selling point of the usability tests is that they

are supposed to show real users during real use cases. Instead the use of stock videos from previous tests was used during the sales meetings and the production of pitch videos stopped. It did however get me thinking. Why did the videos seem fake? The tasks the users were solving were real enough. They were real use cases. It did however seemed to not be THEIR use cases.

As part of the investigation a pitch test for the website Dell.dk was created with a completely exploratory approach. The two testers were first asked to describe their preferences when shopping electronics online, in an attempt to make them reflect upon them. Lastly they were asked to visit Dell.dk, explore the website and possibly try ordering a product based on their own preference. For this they were given 20 minutes.

While the suggestion to order a product is highly suggestive, considering that the test is supposed to be exploratory, it was added to give the testers a hint to what they could spend their time on. Adding the word "possibly" and putting an emphasis on the fact that the product should be selected based on personal preference made the task as open as possible. During my time as an intern i had encountered different types of testers. While some seem genuinely adventurous and try to relate to the test scneario, others I called "automatic" testers. Automatic testers read their task and complete the task, without trying to reflect upon what is going on. My fear was that the latter type of tester would not be able to perform an exploratory test.

When i eventually received two testvideos from the pitch test, the two testers that had been randomly assigned to perform the test turned out to be one of each type. One tester completely missed the word "possibly" (intentionally or not), picked out a, seemingly, random computer that didn't correspond to her own verbalised preference and ordered it. The other tester went exploring among tablet, printers and AlienWare computers (which he quickly agreed with himself didn't fit his needs).

The findings of the adventurous tester were the ones that ended up being sent to the client advisor and it received great feedback.

Chapter 2

Problemanalysis

Usability testing was invented as a response to the user exclusion of the heuristic evaluation method. Instead, it was argued, we should observe the user while he's using the website and find REAL problems. Most usability testing today consists of testers going through a website using predefined tasks. The problem with predefined tasks however, is that they are predefined. They are constructed by a human being who has made a choice on what he has deemed to best cover the website and discover potential flaws. While the tasks might have been constructed by doing a task analysis or through communication with the developer, they are sitll the product of human choice.

How could this be countered though? One way would be to design standardized test designs with generic questions that cover any use case. Some kinds of website have very generic functions. E-commerce websites will for example always have the ability to search for and buy a product. While standardized test designs will eliminate the human factor and any subjectivity in the tasks it still has a flaw it shares with the standard tasks. All testers will receive the same tasks and therefore they will always follow the same patch and find the same errors (to some extend). Also, while there are some functions should be expected to be present on all websites, many websites try to stand out and it might be in the differences that the errors lie.

So what could be done? My proposal is to make the test exploratory. Instead of imposing a test scenario on the tester, expecting him to relate, and trying to guide the tester to where you THINK he would want to go, it would be easier to ask him; "Where would you go? What would your purpose on this website be?". There are however some early concern regarding the use of this method. I expect, from experience, that not all testers will perform equally well. It requires imagination and the ability to reflect upon your own preferences. I do however find it important that the method can be performed

by everyone, since the context in which it is designed to be used in is where the testers will be pulled from a large databate, as to make it as random as possible. It might not be a bad thing that testers aren't able to explore the website and instead REVERT to default since it might lead them in a testing path that the others won't cover.

2.1 Problemstatement

With this in mind the problemstatement of this thesis is as follows:

> **When performing remote asynchronous usability testing, is there a difference in using a tasksbased or exploratory approach as a walk-through method?**

To answer the problemstatement, the following research questions are posed:

- Is there a (significant) difference in the testers subjective impression and opinion of the website with each method?
- Is there a difference in the type and amount of errors found with each method?
- Is there a difference in the individual testers ability to use the exploratory walk-through method?
- Is it possible to categorize who better can use the exploratory walkthrough method?

Chapter 3

Method

3.1 Heuristic Evaluation

In the late 80s and early 90s the Heuristic Evaluation method became very popular. Heuristic Evaluation is a formal method where one or more evaluators go through an interface design to find as many usability problems as possible. Most attribute Nielsen to be the author of the method, but it was widely used before. At the time however there was a huge collection of usability guidelines that had *"on the order of one thousand rules to follow"* (Nielsen and Molich, p. 249) and due to this being intimidating to developers, Nielsen and Molich argued that *"most people probably perform heuristic evaluation on the basis of their own intuition and common sense instead"* (Nielsen and Molich, p. 249). To reduce the complexity the *nine basic usability principles* were proposed:

1. Simple and natural dialogue - *Dialogue between the user and the system should not contain irrelevant or rarely needed information.*

2. Speak the user's language - *The system needs to communicate in a language that's understandable by the user (not in code!).*

3. Minimize user memory load - *The user should not unnecessarily have to remember information.*

4. Be consistent - *Follow platform conventions. Present the same information, the same way throughput the system.*

5. Provide feedback - *Make sure the user always knows the current state of the system.*

6. Provide clearly marked exits - *Provide the user with a clearly marked way to undo errors or ways of exiting the current state (e.g. homepage button on websites).*

7. Provide shortcuts - *Provide accelerators in the form of shortcuts for expert users.*

8. Good error messages - *Don't just tell the user he made an error, tell him WHAT that error was.*

9. Prevent errors - *Avoid putting the user in a situation where he can make an error in the first place.*

   The principles were later revisited by Nielsen in 1993 (18) and a tenth point was added:

10. Help and documentation - *Provide the user with fast and effective ways to seek help if needed.*

The ten item list is today known as *Nielsen's Heuristics* and is still in use to this day.

With the increased focus on lowering the cost and expand testing of interfaces to not only be a tool used by big companies, alternatives were investigated. In 1992 Nielsen performed a study (16) comparing usability specialists, non-specialists and what he called double experts. The non-specialists consisted of 31 computer science students who were novices when it came to usability, but not with computers. The 19 usability specialists were defined as "people with a graduate degree and/or several years of job experience in the usability area". Lastly the double specialists were 14 people who were both usability experts and had special expertise in the type of interface that was tested. In the study this was a voice response system. The study showed, not surprisingly, that the double specialists performed the best of all, while the regular specialists performed better than the non-specialists. Nielsen however talks about trade-off analysis. While the study showed that the double specialists performed the best, the cost to benefit ratio might be better when using another group. If one double specialist costs the same as three regular specialists then according to the findings illustrated in figure 3.1 one would be better off hiring two or three regular specialists.

3.2 Usability testing

While the Heuristic Evaluation had focus on 'you need to be an expert to know what is wrong' Usability Testing includes the observation of the users in the evaluation. The idea is to observe a user working with the system and then note down errors and base assessment of the usability. The classical way to achieve this was to ask users to solve

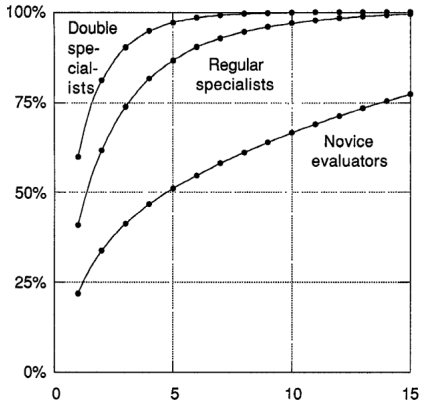


Figure 3.1: Nielsen found that one double specialist is as effective as two regular specialists or 8-9 novice evaluators. (16, p. 377)

tasks while making use of the think-aloud protocol in a laboratory setting (Rubin). A drawback however turned out to be the increase in cost of time and resources, compared to Heuristic Evaluation. A comparative study done in 1991 (Jeffries et al.), comparing four user interface evaluation techniques, found that, while Heuristic Evaluation requires several experts in the field, it had lower cost than Usability Testing while identifying more problems. The search was on develop a cost-effective way of doing usability testing that could compete withe Heuristic Evaluation.

3.3 Remote Usability Testing

One of the first mention of doing usability testing remotely was made some 20 years ago with an empirical study by Hartson et al. Here it was defined to be *"usability evaluation, wherein the evaluator, performing observation and analysis, is separated in space and/or time from the user"* (Hartson et al., p. 228). The separation in space was named remote synchronous usability testing, while the separation in both time and space was named remote asynchronous usability testing

The main reason for doing remote usability testing is to easily be able to include a broader group of testers, especially geographically, while saving time by having the testers perform the test at home instead of travelling to a usability lab. The remote synchronous usability test however suffers from being almost as resource demanding as a traditional lab test since the separation only is spatial, meaning that the evaluator still is required to be present in real time to monitor and control the test (Dray and Siegel).

3.4 Remote Asynchronous Usability Testing

Contrary to the remote synchronous method, the remote asynchronous method is far less resource demanding due to the both spatial and temporal separation of the tester and evaluator. This does however require that the tester is able to perform the test on his own, without the support of the evaluator. Early studies on the remote asynchronous method, made use of extensive training of the users with the user and instructor meeting physically (Hix et al.; Castillo et al.; Thompson).This however contradicts the whole purpose of the method. The extensive training was needed due to the focus being on user reported errors. If the testers are only required to complete tasks, while the error identification is being made through review of videorecording, the tester will require minimum training. The training can then be performed online instead.

3.5 Walkthrough Method

When set on the testing method, one also need to decide on the walkthough method of the tester. The classical approach is to present the tester with a series of tasks that are representative of the use of the website. For example, to find and buy a product on an e-commerce website.

Another walkthrough method is with exploratory test tasks. This relies on the testers ability to make the website relatable to himself, in exchance for more sincere results.

3.5.1 Test Tasks & Exploratory Test Tasks

The use of test tasks is classic in usability testing. The tasks are usually designed based on a product identity statemen or based on a tasks analysis. Most importantly the tasks should be as representative as possible of its the systems or websites uses. However, in 2001 Richard E. Cordes voiced concerns about bias in task-selection and it being overlooked (Cordes). One of the sources of bias is that the evaluator might have a tendency to focus on product areas that were important or controversial during development or important to the developer. As a supplement to the tradional tasks, Cordes introduces the use of User-Defined Tasks, which are tasks designed by the tester themselves during the usability testing session.

While the concerns of Cordes applied to standard usability lab-based testing, Bruun and Stage (Bruun and Stage) argue that in the context of remote asynchrnous testing *"predefined tasks compromise validity, because users are forced into artificial usage situ-*

ations"(Bruun and Stage, p. 2117). Alternatively, they argue, users can work on their own authentic tasks. In the same study they concluded that "*users solving predefined tasks identified significantly more usability problems with a significanly higher level of agreement than those working on their own authentic tasks*" (Bruun and Stage, p. 2117). However the usability problems were self-reported and identified over a timeframe of four weeks during daily use. It could be argued that self-reporting doesn't support such a long timeframe, since the testers might not always be in "error-finding"-mode, especially not when the context is daily use.

The research in the area of remote asynchronous usability testing, seems to be focused on either the use of user-reports (Hartson and Castillo; Hartson et al.; Hix et al.) or auto data logging (Millen; 25; 24; Scholtz and Downey) for error identification. The method of doing review of video recordings to detect errors, doesn't seem to have been brought over with the shift from usability testing to remote asynchronous usability testing. While the exploratory walkthrough method has been proposed by some through the years, but with in self report context.

Merging the review of video recordings as error identification, with remote asynchronous usability testing using an exploratory walkthrough method seems to be an area yet to be explored.

3.6 Data Collection

There are many different methods and techniques used to collect data from a testing session. Some collect data automatically during the session (click stream, task time, eye traching, bio data) while others are performed by the user himself (self logging, diary, questionnaire) This section will only cover the ones being used in this study.

3.6.1 Thinking-aloud Protocol

A technique often used in Usability Testing to gather data is the thinking-aloud protocol. The use of verbal reports as data was first proposed by Ericsson and Simon in their seminal work Protocol Analysis: Verbal Reports as Data (Anders and Simon). In 2000 'Thinking Aloud: Reconciling Theory and Practice' (Boren and Ramey) found that, even though Ericsson and Simon usually were the primary source cited in works using the thinking-aloud protocol, practice didn't follow theory. While the classical approach dictated total silence from the evaluator besides giving the tester reminders to think-aloud, the norm is now for evaluators to ask for elaborations.

The thinking-aloud protocol is just as effective at when used without evaluators (Bonde), which makes it a great tool for data collection when doing remote asynchronous usability testing.

3.6.2 Poststudy Questionnaires

A popular way of gathering subjectic usability metrics is by conducting poststudy questionnaires. Usually these questionnaires consist of a range of statements where the test user needs to acknowledge their amount of agreement on a likert scale. The amount of statements and points in the likert scale differs from each method. In a study done by (Tullis and Stetson) the sensitivity of five poststudy questionnaires were compared. The fine questionnaires were:

- SUS (10 questions using a 5 point likert scale)
- QUIS (27 questions using a 10 point likert scale)
- CSUQ (19 questions using a 7 point likert scale)
- Words (Testers pick words from a list)
- Ours (Tullis and Stetsons own questionnaire, 9 questions using a 7 point likert scale)

The sample size of 123 was randomly assigned one of the 5 methods to evaluate two websites after completing two tasks on each site. The samples were then randomly selected in subsamples of 6, 8, 10, 12 and 14 for each method. Each subsample was then used to determine how fast each method would reach the correct conclusion. As shown in figure 3.2 the SUS was fastests to reach 100% correct conclusions at a sample size of 12. Overall SUS performed superior to all the poststudy questionnaires tested.

System Usability Scale (SUS)

The System Usability Scale (Brooke) consists of 10 questions suing a 5 point likert scale.

Sauro (Sauro) gathered data from a series of papers and articles on SUS suchs as (Tullis and Stetson; Bangor et al.; Sauro). The analysis included in total 446 surveys and usability studies. Sauro then categorized the studies by the type of interface tested and found the benchmark for each. In total 8 different categories were created, as well as a global benchmark as shown in figure 3.3. Interestingly enough, the global mean of SUS-score isn't 50, but instead 68. This means that the SUS-score is slightly inflated. For

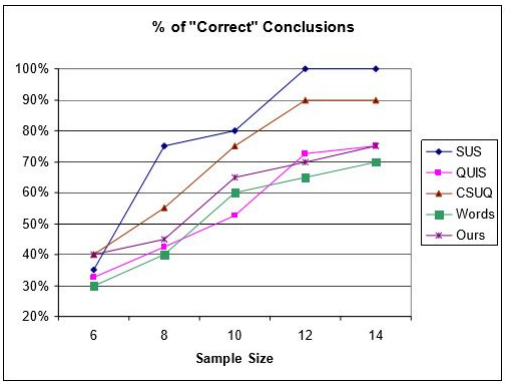


Figure 3.2: The sensitivity of five poststudy questionnaires was tested. SUS was the fastests to reach 100% correct conclusions at a sample size of 12 (Tullis and Stetson).

example receiving a SUS-score of 60 doesn't mean that you are better than 60% of the websites out there. Instead you're only better than 29%.

Table 8.7 SUS Benchmarks by Interface Type

Category	Description	Mean	SD	*N*	99% Confidence Interval	
					Lower Limit	Upper Limit
Global	Data from the entire set of 446 surveys/studies	68	12.5	446	66.5	69.5
B2B	Enterprise software application such as accounting, HR, CRM, and order-management systems	67.6	9.2	30	63	72.2
B2C	Public facing mass-market consumer software such as office applications, graphics applications, and personal finance software	74	7.1	19	69.3	78.7
Web	Public facing large-scale websites (airlines, rental cars, retailers, financial service) and intranets	67	13.4	174	64.4	69.6
Cell	Cell phone equipment	64.7	9.8	20	58.4	71
HW	Hardware such as phones, modems, and Ethernet cards	71.3	11.1	26	65.2	77.4
Internal SW	Internal productivity software such as customer service and network operations applications	76.7	8.8	21	71.2	82.2
IVR	Interactive voice response (IVR) systems, both phone and speech based	79.9	7.6	22	75.3	84.5
Web/IVR	A combination of Web-based and interactive voice response systems	59.2	5.5	4	43.1	75.3

Figure 3.3: The data from 446 surveys and usability studies were used to create SUS benchmarks for each interface type (Sauro and Lewis).

3.7 Sample Size

Sample size has sparked neverending discussion in the field of usability testing. Both in regards to the amount of test users and evaluators used. The sample size however depends on if the study being performed is summative and formative. Summative studies makes use of measurement-based evaluation while formative studies focuses on the detection and elimination of usability problems. In this study I will be looking at both summative

evaluation in the form of a comparison of means from a System Usability Scale and I'll be comparing the amount of errors found for each method. In the following subsections I will be calculating the optimal sample size for this study.

3.7.1 Sample Size: Formative Evaluation

Formative studies concentrate on finding usability problems. The question usually is how big a portion of the total amount of usability problem present, one is satisfied to uncover. The detection of usability problems isn't directly proportional to the sample size used since some usability problems will be harder to uncover than others. In the early 90's however a series of papers (Virzi; Nielsen and Landauer), proposed that the finding of usability problems, in relation to the amount of test users or evaluators, had a distribution that closely resembled that of a Poisson distribution. Therefore one could approximate the amount of test subjects needed to find at least a set percent of all usability problems present using the following equation:

$$Found(i) = N(1 - (1 - \lambda)^i) \tag{3.1}$$

where

N is the total number of usability problems

λ is the probability a test person has of finding a new usability problem, meaning that $1 - \lambda$ is the probability of a usability problem remaining unfound

i is the number of test users

$Found(i)$ is the amount of usability problems that have been found at least once by i number of test users

To plot the graph of the equation one needs to first calculate λ. This is done by looking at the average amount of usability problems found by each test user in relation to the total amount of problems found. This calculation was done by Nielsen and Landauer (Nielsen and Landauer) while going through 11 different studies. They calculated λ to be 0.31, meaning that each test users on average found 31% of the total amount of usability problems. If one then had to approximate the amount of usability problems found using 5 test users going through a system that contains a total of 15 problems, the equation would look like this:

$$Found(i) = N(1 - (1 - \lambda)^i) \tag{3.2}$$

$$Found(i) = 15(1 - (1 - 0.31)^5) \tag{3.3}$$

$$Found(i) = 15(1 - (0.69)^5) \tag{3.4}$$

$$Found(i) = 15(1 - (0.69)^5) \tag{3.5}$$

$$Found(i) = 15(1 - 0.156) \tag{3.6}$$

$$Found(i) = 15(0.844) \tag{3.7}$$

$$Found(i) = 12.66 \tag{3.8}$$

The total percentage of usability problems found is calcualted in the parentheses in equation 3.7 (0.844 = 84.4%) while the approximated amount of usability problems is found in equation 3.8 to be 12.66. The curve of the equation when plotted can be seen in figure 3.4.

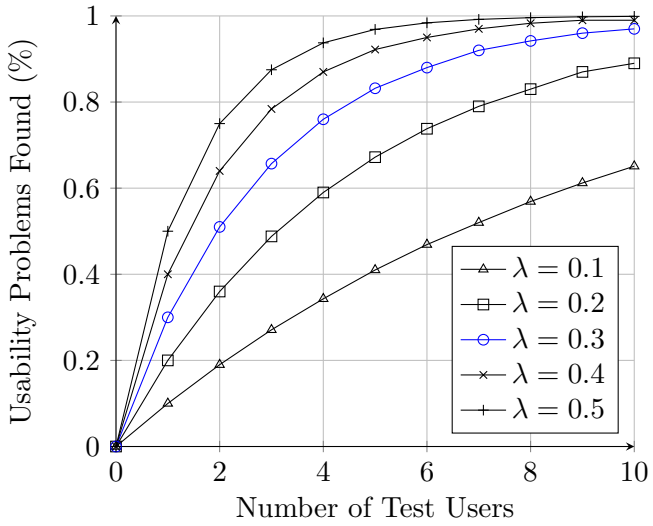


Figure 3.4: With a detection rate λ of 10%, 20%, 30%, 40% and 50% the curve looks as follows.

As figure 3.4 shows, at sample size 5 the amount of usability problem found is just above 80% (83.2%) while it rises to 88%, 92%, 94.2%, 96% and finally 97% at sample

size 10. Using this as an argument the mantra of 'five testers is enough' gained ground. One major drawback of using this mathematical model to accept a sample size of 5 for all usability studies however is that it automatically assumes that all errors have the same detection rate λ. As figure 3.4 illustrates, at a detection rate λ of 0.2 one needs to use a sample size of 8 to discover the same amount of usability problems. We can then conclude that the formative part of the study requires a sample size of at least 5 to reach 80% usability problems found. However, an increase of sample size to 8 will cause this number to reach 94%, while the lower detection rate 0.2 will pass the 80% mark (from 68% at 5). While 5 might be the minimum, 8 seems more comfortable.

3.7.2 Sample Size: Summative evaluation

Summative studies usually are based on quantitave data collected using methods such as satisfaction ratings, completion rates and task time. Using this data one can calculate means which can be used to compare with industry standard and benchmarks. It can also be used to compare with earlier studies to see improvements and to do hypothesis testing. Because of its conclusive nature, summative usability studies resemble classical scientific studies. In this study I'll be letting each test user fill out a post-test questionnaire in the form of a System Usability Score which was previously introduced in section 3.6.2. I'll then be able set up an alternative and null-hypothesis and either confirm or reject this by testing if there is a significant difference between the means calculated from the system usability score.

Based on the data from both the SUS-score agreements mentioned in section 3.6.2 and the amount of problems found mentioned in the previous section the following table can be made:

Sample size	**SUS**	$\lambda = 0.2$	$\lambda = 0.3$
5	N/A	68%	83%
6	35%	74%	88%
8	75%	83%	94%
10	80%	89%	97%

Table 3.1: The approximated amount of usability problems found aswell as SUS effectiveness at each sample size.

Determining the sample size of the study is done by doing a trade-off analysis. What is the smallest sample size possible with the highest relative gain, while still maintaining an acceptable confidence level.

While $\lambda = 0.3$ already has a detection rate of 83% at 5 samples, the SUS-score requires a higher sample due to its quantative nature. The jump from 6 to 8 sample size is 40% while the jump from 8 to 10 is a mere 5%.

A sample size of 8 seems reasonable for this study, based on the problem detection rates at 83% ($\lambda = 0.2$), 94% ($\lambda = 0.3$) and a SUS-score effectiveness of 75%.

3.8 Hypothesis Testing

As part of the summative evaluation of the website using the SUS-score, i will be performing hypothesis testing to answer one of the research questions posed in chapter 2:

> **Is there a (significant) difference in the testers subjective impression and opinion of the website with each method?**

To answer the research question the null-hypothesis will be

H_0 = There is no significant ($\alpha = 0.05$) difference between the two SUS-score means.

and the alternative hypothesis

H_1 = There is a significant ($\alpha = 0.05$) difference between the two SUS-score means.

The alpha (α) value in the hypotheses signifies how likely one wants to risk making a Type I error, a wrongful rejection of the null-hypothesis. With $\alpha = 0.05$ there is a 5% risk, meaning that, over the long run, there should only be a Type I error in one out of 20 tests. But why not set $\alpha = 0.01$ then? Or even $\alpha = 0.001$? Surely one would rather make a Type I error every 1000 test instead. When α decreases the risk of making a Type II error, denoted by the beta (β) value, instead increases. The only way to reduce both error rates is to increase the sampe size.

To test the hypothesis one can use a Students t-test. This study will be comparing two different methods, using different testers for each method. The test will therefore have two samples that are independent of eachother. Because of this an independent two sample

t-test will be suitable to hypothesis test the two means. To find the p-value one needs to first calculate t using the following equation

$$t = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{3.9}$$

where

$\bar{x_1}$ and $\bar{x_2}$ are the means of sample 1 and sample 2

s_1 and s_2 are the standard deviation of sample 1 and sample 2

n_1 and n_2 are the sample size of sample 1 and sample 2

When t is found one can use the degrees of freedom $(n_1 + n_2 - 2)$ to look up the p-value in a t-table. If $p > \alpha$ then we can't reliably reject H_0 and therefore not accept H_1.

Chapter 4

Pretest

The test will consist of two groups of 8 people testing the same website, each using a different walkthrough method. One group will be using classical test tasks, while the other will use an exploratory walkthrough method.

4.1 Choice of test website

The choice of website the two methods will be tested on is important. Ideally a new website should be developed for the purpose of testing alone. This would make it possible to intentionally incorporate a set amount of errors throughout the website. If the number of errors a website contains is known, the actual percentage of errors found per method could be calculated. It is however not feasible to do so, due to the time constraints this thesis contains. It might also be challenging to limit the amount of errors the website contains, to only be intentional.

Due to this, an existing site will be used. The website will need to meet the following requirements:

1. The website should not be too specialized - *E.g. www.rytterhjoernet.dk. This is especially important for the exploratory testers since they need to be able to relate to the website.*

2. The website should not be too simple - *An extreme example of this is www.ugedag.dk. The website needs to be complex enough to support creation and completion of tasks.*

3. The website should not be too popular - *E.g. Facebook.com. The testers need to ideally not be familiar with the websites they are testing, to prevent them from doing things by routime*

4. The website should not be too 'good' - *If a website has few errors and a high SUS-score the distinction (or lack of) between the two groups will be hard to measure and prove.*

To produce a list of potential test websites, it was easiest to go by the aforementioned 4th requirement. The website Trustpilot has a list of websites rated by the users. While the ratings aren't a direct indication of how 'good' or usable a website is, it is a good indicator for picking out websites that might. In the end the choice came upon the website Trendway.dk. As seen in figure 4.1 Trendway.dk had mostly received negative reviews. However most of these were due to delayed products and not the usability of the website itself.

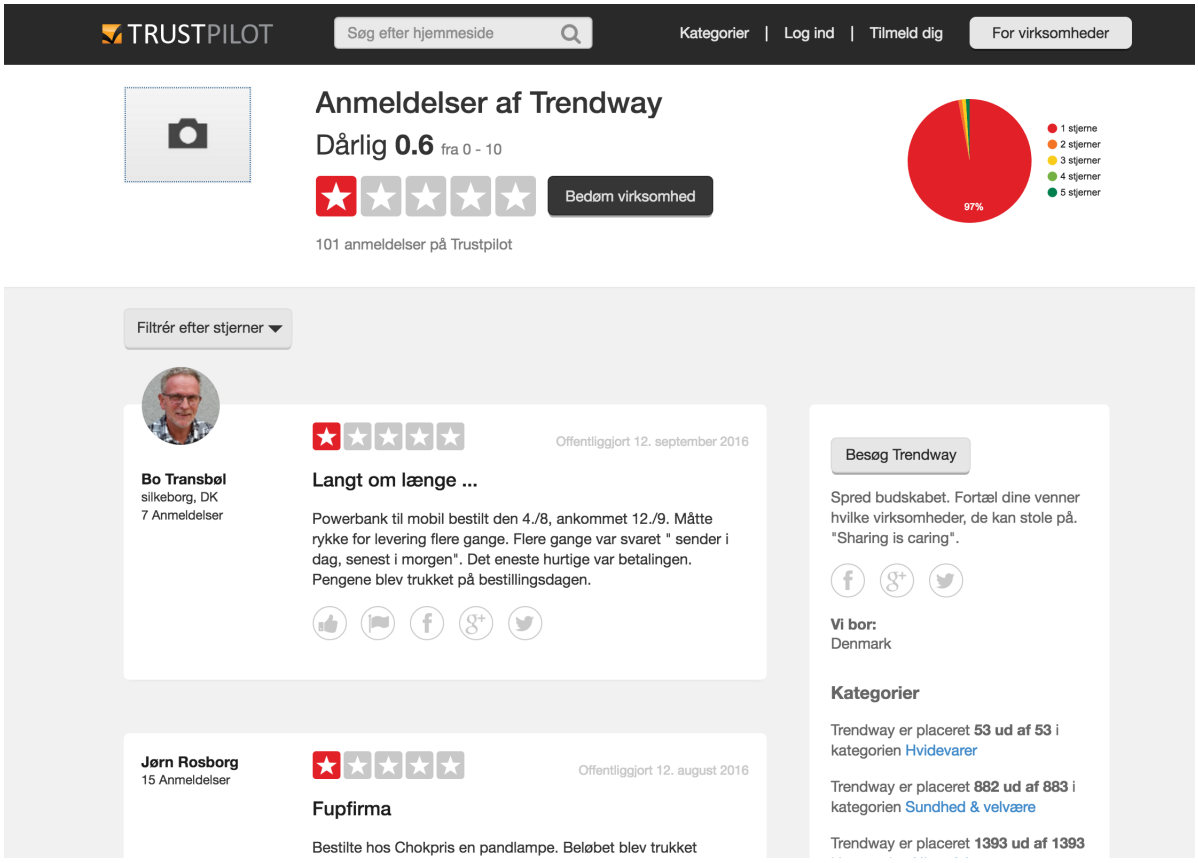


Figure 4.1: The page of Trendway.dk on Trustpilot. As shown the website had mostly received negative reviews. Most of these mentioned delayed products.

When visiting the website though it became clear that it contained a high amount of errors located in many different areas. This made it perfect for testing purposes, since any difference in errors found or SUS-score would be more prominent.

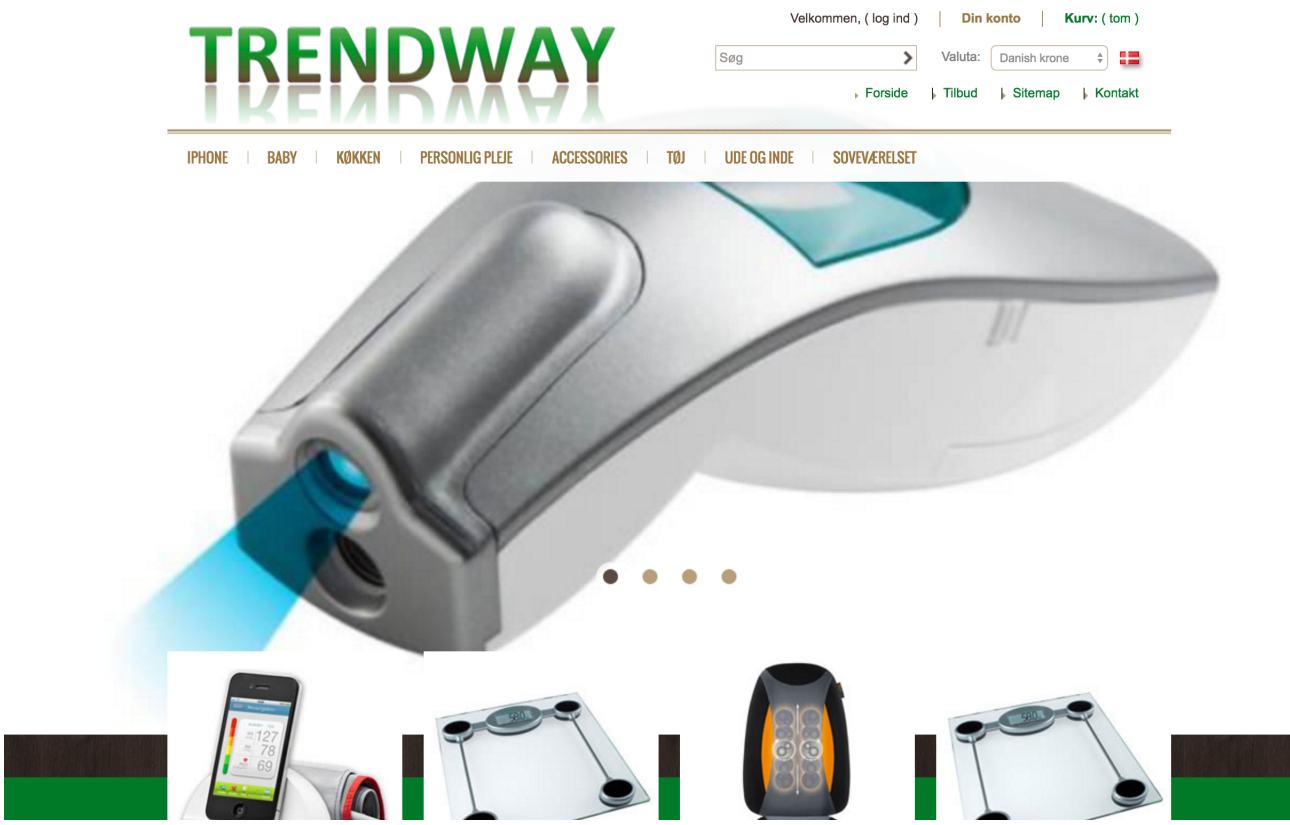


Figure 4.2: The frontpage of Trendway.dk as of 18.10.2016. When going through the website it was clear that it contained a high amount of errors, which made it perfect for testing.

4.2 Test Tasks

4.2.1 Taskbased Test

There are many ways to design tasks for a test. Most importantly the tasks need to reflect the uses of the website. This can be done doing a task analysis or based on a product identity statement, which lists the intended uses of the products (17, p. 185).

Normally the tasks would have been created through extensive communication with the owner or developer of the website, to base the tasks upon his focus and wishes. For example some companies might experience a high shopping cart abandonment rate and wants to make sure that the testers go through a checkout flow. This involvement wasn't possible in this case. Instead i chose to do a task analysis of the website inspired by the Noun and Verb Identification Technique commonly used in object-oriented programming (Holmes and Joyce, p. 109), to identify the classes and functions of a newly started program. The Noun and Verb Identification Technique requires the programmer to first

describe the problem or task the program needs to solve and then identify all the nouns and verbs.

In the same way it is possible to create possible tasks for the users by going through a website while describing the available actions as seen below:

> Trendway er en indkøbsportal for man kan købe en lang række varer. Der er adgang til en konto der tilgåes ved at oprette en bruger. Der er en indkøbskurv hvori der kan lægges varer. Disse varer kan tilføjes og fjernes kurven og antallet kan ændres. Fra indkøbskurven kan man bestille varen til afsending. Varerne er placeret under en række kategorier. Under hver kategori kan varerne sorteres. Priserne kan vises i forskellige valutaer. På hver side kan et søgefelt tilgåes hvor der kan søges efter varer. Der kan tilgåes et sitemap, en tilbuds side og kontakt. Nederst findes der under kategories Information link til kontaktformular, handelsbetingelser, kundeoplysninger (virker ikke), om os (virker ikke). Under kategorien Vores Tilbud findes links omhandlende tilbud (virker ikke). Under kategories Din Konto kan den oprettede konto tilgåes, kontooplysninger kan ændres, fakturerings- og leveringsadresser kan tilføjes, rabatkuponer kan ses (der er ingen) og odrehistorik kan tilgåes.

When the discription is done any action described is identified and bolded out. Obvious overall actions are left out such as "købe en lang række forskellige varer" and actions that can't be performed due to links not working etc. are not included.

> Trendway er en indkøbsportal for man kan købe en lang række forskellige varer. Der er adgang til en konto der tilgåes ved at **oprette en bruger**. Der er en **indkøbskurv** hvori der kan **lægges varer**. Disse varer kan **tilføjes og fjernes kurven** og **antallet kan ændres**. Fra indkøbskurven kan man **bestille varen til afsending**. Varerne er placeret under en række kategorier. Under hver kategori kan **varerne sorteres**. Priserne kan vises i forskellige valutaer. På hver side kan et søgefelt tilgåes hvor der kan **søges efter varer**. Der kan tilgåes et sitemap, en tilbuds side og kontakt. **Nederst findes** der under kategories Information link til **kontaktformular**, **handelsbetingelser**, kundeoplysninger (virker ikke), om os (virker ikke). Under kategorien Vores Tilbud findes links omhandlende tilbud (virker ikke). Under kategories Din Konto kan den oprettede konto tilgåes, **kontooplysninger kan ændres**, **fakturerings- og leveringsadresser kan tilføjes**, rabatkuponer kan ses (der er ingen) og **odrehistorik kan tilgåes**.

After identifying the actions, the words are then written out in sentences that explain the action.

1. Oprette en bruger
2. Indkøbskurv (kan der) lægges varer
3. Varer (...) tilføjes og fjernes (fra) kurven
4. Bestille varen til afsending
5. Varerne (kan) sorteres
6. Søges efter varer
7. Nederst findes (link til) kontaktformular
8. Nederst findes (link til) handelsbetingelser
9. Kontooplysninger kan ændres
10. Fakturerings- og leveringsadresser kan tilføjes
11. Orehistorik kan tilgåes

The actions are then placed in 3 categories; **possible**, **leading** and **not possible**. The first category covers all actions that are possible to be integrated as a test task. The second category covers actions that are possible, but might need too much guidance from the test tasks. Actions such as the use of filters can not be an independent test task in itself, but might be performed by the tester on his own initiative while solving other tasks. The last category covers actions that are not possible. This covers actions such as purchasing a product.

Possible	**Leading**	**Not Possible**
1, 2, 3, 7, 8	5, 6, 9, 10	4, 11

Table 4.1: After identifying the actions they were divided into 3 categories: Possible, Leading and Not Possible.

After categorising the actions they are put in a approximated chronological order. For example, finding a product will happen before adding the product to the shopping cart and the users will usually wait as long as possible before creating an account.

1. Find a product based on a preference (6, 5)

2. Add the product to your shopping cart (2, 3)

3. Create an account using fake informations (1)

4. How do you contact customerservice? (7)

5. What is the right of cancellation on the site? (8)

Lastly a few test tasks are added, asking for opinion on the site and if the user would feel safe shopping here. To make the tasks more relatable, they are linked to an overall testscenario:

> **Testscenario** *You have to buy a birthday gift for a friend. You've heard that the website www.trendway.dk has a big selection of various products and therefore chooses to visit it.*

To make the **Find a product based on a preference** task relable to the testscenario it was changed to **Find one or more birthday gifts for your friend and add it/them to the basket. You have decided to spend a maximum of 800kr.**.

The final task design can be seen in Appendix B (danish) and Appendix C (english).

4.2.2 Exploratory Test

The exploratory test design looks very similar to the one used in the pitch test mentioned in section 1.1 with a few additions.

It is important that the testscenario properly introduces the exploratory walkthrough method to the testers, since it hasn't been presented to them before. It is also important to make sure that the tester understands that the duration of the test (20-30 minutes) still is the same as with all other tests. Lastly a reminder is added to make sure that the tester remembers to think-aloud.

The exploratory test design consist of three parts:

1. **Introspection** (5 mins.)

   The introspection part is the same as it was in the pitch test. It is designed to get the tester to verbalise his needs on the subject, before knowing what website he will be testing. Doing this, he might more readily know his own preferences and needs while testing.

2. **Testing** (16 mins.)

   Here the tester is presented to the website he needs to test. He is also reminded to

go by his own interests and be thorough. Lastly another quick reminder is added to make sure that the tester remembers to think-aloud.

3. **Reflection** (5 mins)

   This part is new compared to the pitch test. The tester is asked to state his opinion on the exploratory approach and how it compares to the task-based approach. Lastly he is asked if he has a preference. This part was added to explore the testers opinion on the approach and to see if there is any correlation between preference and how well the tester performs.

The final exploratory task design can be found in Appendix D (danish) and Appendix E (english).

4.3 Testers

The two groups of testers will be randomly picked through UserTribes tester database. Each group will have an equal amount of men and women, as well as a broad distribution of age and experience (amount of tests previously completed). Every tester in the database has been introduced to thinking-aloud while testing as well as completing tasks and has as a minimum of one approved testvideo to prove they are able to do it. Due to this, the testers have an understanding of what is required of them.

4.4 Test Budget

The testers are paid 100kr which is the minimum payment for online testers at UserTribe. While the use of voluntary tester often is used in the academic world, this option did not seem appropriate in this study. Since the testers are drawn from UserTribes own testerpanel, the range of age and experience is gonna be more evenly spread. This will most probably not be the case if the testers are recruited through the authors social network which would have been the case, had the test been conducted with voluntary testers.

Chapter 5

Results

A high amount of different data was collected from the two tests to answer the research questions posed in section 2.1. The following list provides an overview over what research question each section of the chapter seeks to answer.

- **Errors**
  Is there a difference in the type and amount of errors found with each method?

- **System Usability Scale**
  Is there a (significant) difference in the testers subjective impression and opinion of the website with each method?

- **Preferences**
  Is there a difference in the individual testers ability to use the exploratory walkthrough method?
  Is it possible to categorize who better can use the exploratory walkthrough method?

5.1 Errors

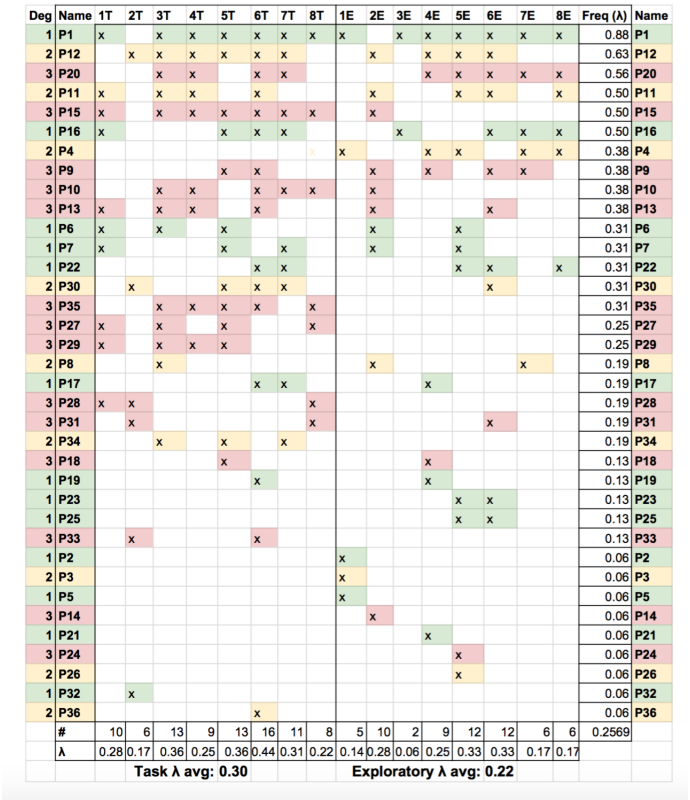
A total of 36 errors were found among the 16 testers. The 36 errors were ordered in 3 categories and given a color based on the severity:

1. Cosmetic or minor errors (Green)

2. Serious errors (Yellow)

3. Critical errors (Red)

The cosmetic/minor category covers errors that are minor nuisances such as pictures not loading (P1) or translation errors (P21). Serious errors cover errors that cause disruption, but doesn't prevent actions. Examples of serious errors are lack of information on product (P11) or lack of a sorting feature (P30). Lastly, the critical errors are high priority errors that either completely prevent actions or otherwise highly disturb the users experience. Examples of this are inability to choose a size/color of a product (P10), having to enter adress information twice during check out flow (P15) and confusing/overwhelming error messages (P28). An overview and description of each error can be found in appendix I together with a series of screencaps despicting some of the errors, in appendix J.



Deg	Name	1T	2T	3T	4T	5T	6T	7T	8T	1E	2E	3E	4E	5E	6E	7E	8E	Freq (λ)	Name
1	P1	x		x	x	x	x	x	x	x		x	x	x	x	x	x	0.88	P1
2	P12		x	x	x	x	x	x			x		x	x	x			0.63	P12
3	P20			x	x		x	x					x	x	x	x	x	0.56	P20
2	P11	x		x	x		x				x			x	x		x	0.50	P11
3	P15	x		x	x	x	x	x	x		x							0.50	P15
1	P16	x				x	x	x				x			x	x	x	0.50	P16
2	P4									x			x	x		x	x	0.38	P4
3	P9					x	x				x		x		x	x		0.38	P9
3	P10			x	x		x	x	x		x							0.38	P10
3	P13	x		x	x		x				x				x			0.38	P13
1	P6	x		x		x					x			x				0.31	P6
1	P7	x				x		x			x			x				0.31	P7
1	P22						x	x						x	x		x	0.31	P22
2	P30		x			x	x	x							x			0.31	P30
3	P35			x	x	x	x		x									0.31	P35
3	P27	x		x		x			x									0.25	P27
3	P29	x		x	x	x												0.25	P29
2	P8			x							x					x		0.19	P8
1	P17						x	x					x					0.19	P17
3	P28	x	x						x									0.19	P28
3	P31		x						x						x			0.19	P31
2	P34			x		x		x										0.19	P34
3	P18					x							x					0.13	P18
1	P19						x						x					0.13	P19
1	P23													x	x			0.13	P23
1	P25													x	x			0.13	P25
3	P33		x				x											0.13	P33
1	P2									x								0.06	P2
2	P3									x								0.06	P3
1	P5									x								0.06	P5
3	P14										x							0.06	P14
1	P21												x					0.06	P21
3	P24													x				0.06	P24
2	P26													x				0.06	P26
1	P32		x															0.06	P32
2	P36						x											0.06	P36
	#	10	6	13	9	13	16	11	8	5	10	2	9	12	12	6	6	0.2569	
	λ	0.28	0.17	0.36	0.25	0.36	0.44	0.31	0.22	0.14	0.28	0.06	0.25	0.33	0.33	0.17	0.17		
		Task λ avg: 0.30								Exploratory λ avg: 0.22									

Figure 5.1: The errors were ordered in three categories, color coded from minor (green), serious (yellow) to critical (red). A bigger version can be found in appendix G.

In section 2.1 the following research question was asked:

Is there a difference in the type and amount of errors found with each method?

To test if there is any significant different between the amount of each type of errors found, it is possible to use Fisher's exact test.

Table 5.1 shows the results of the Fisher's exact test used to compare the overall, critical, serious and cosmetic errors. The p-value shows that there is no significant ($p > \alpha$)

	Task	Exploratory	Total	p
Overall	26	28	36	**0.786**
Critical	12	9	14	**0.3845**
Serious	6	7	9	**1.000**
Cosmetic	8	12	13	**0.16**

Table 5.1: Using a Fisher's exact test the two methods were compared to show if there was a significant difference in the number of errors found in each type.

difference between the errors found using both methods. In other words we can conclude that **there is no significant difference in the amount of errors of each type found with each method.**

The research question did however also question if there was a difference between the type of errors found. While this can't be significance tested, it is still possible to look at tendencies. Figure 5.1 (appendix G) provides a good overview over what errors were found by one method alone. Especially the red cluster consisting of P27, P29 and P35. These 3 errors were exclusively (and to a high extend) identified by the task method group. P29 and P35 both happened relatively late in the check-out process. Since the task method group had been tasked to follow the check-out process until the end, 50% and 63% encountered these two errors. Likewise, another check-out process related error, P15, was encountered by only 1 from the exploratory method group, while all bar one identified it in the task method group. P28 is also exclusive to the task method group and was encountered by the testers during the check-out process. There seems to be a pattern on when the uniquely found errors occur in the shopping process. The errors can be split into four categories signifying this: browsing, shopping, buying and check-out. The browsing category covers anything that doesn't involve searching for or buying a product. This could be no information on handling costs in the terms (P14) or a dead link on the front page (P27). Shopping covers whatever happens while browsing for products. This includes filtering causing an error (P24) or product pictures being too small (P23). Buying is anything that happens when a choice of product has been made to when the check-out begins. This covers anything that happens to the basket, such as it dissapearing (P33) or it not updating when products are removed (P36). Lastly we have the check-out category. Examples have already been covered earlier since, as we found, the task method is heavily overrepresented here. As seen in figure 5.1, if the errors are sorted into categories based on when in the shopping process they are encountered, it becomes evident that all of the

exploratory method group unique finds happen early in the process, while the opposite is true for the task method group.

	Task	Exploratory
Critical	P27, P28, P29, P33, P35	P14
Serious	P34, P36	P4, P26
Cosmetic	P32	P2, P5, P21, P23, P25,

	Task	Exploratory
Browsing	P27	P4, P14, P21, P25, P26
Shopping	P32	P2, P5, P23, P24
Buying	P33, P36	
Check-out	P28, P29, P34, P35	

Figure 5.2: Unique errors found by the exploratory method group happened early in the process, while errors unique to the task method group happened later.

This data is backed by my own observations. Most of the exploratory testers never added a product to the basket or reached the check-out proces. Instead the common reaction to receiving the free rein seemed to be to default into "click-and-tell"-mode. "Click-and-tell"-mode simply means that the tester clicks his way through the website, many times systematically, while reading the text, without relating to what is going on. While this approach might grant some error findings, they are, as can be seen in figure 5.1, mostly cosmetic or minor. In chapter 6 i will be discussing the reason for this and give my recommendations to what could be done to avoid this. For now though, my observations are that **there is a difference in the type of errors found with each method.**

5.1.1 Detection and frequency rate (λ)

As shown in figure 5.1 the detection rate of each tester, the average detection rate of each group, as well as the frequency rate of each error was calculated.

While we found in the previous section that the task method group identified fewer errors than the exploratory method group, they still had a higher average detection rate as seen in 5.1. The task method group had an average of λ 0.30 while the exploratory method group had an average of λ 0.22. While it might not seem logical that the group with the lower detection rate identified a higher amount of error, it implies that the task method testers had a higher detection rate on the fewer problems they identified, while the exploratory method testers had a lower agreement rate.

As discussed in section 3.7.1 Nielsen and Landauer (REF) calculated λ to be 0.31, but the sample size was set to accommodate a λ as low as 0.2. Now that the detection rate and amount of errors had been found, it is possible to calculate the total assumed amount

of errors.

$$Found(i) = N(1 - (1 - \lambda)^i) \tag{5.1}$$

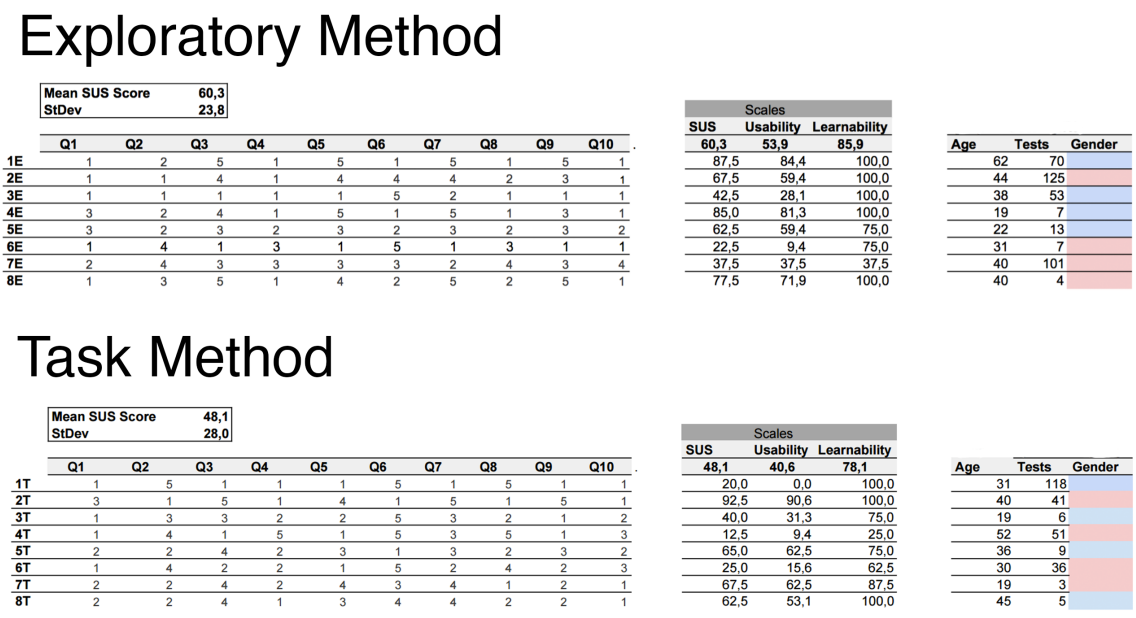
Inserting the total amount of errors found (36), the sample size (16) and the average detection rate over both groups (0.257), we can solve for N:

$$\begin{aligned} 36 &= N(1 - (1 - 0.257)^{16}) \\ 36 &= N(0.99) \\ N &= 36.31 \end{aligned} \tag{5.2}$$

N needs to be an integer since there are no "1/3 errors". The total amount of errors present is therefore assumed to be **36**.

5.2 System Usability Scale

The website acheived SUS-score on respectively **60.3** for the exploratory method and **50.3** for the task method. The results can be seen in figure 5.2 and in appendix H.



Exploratory Method

Mean SUS Score	60,3
StDev	23,8

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1E	1	2	5	1	5	1	5	1	5	1
2E	1	1	4	1	4	4	4	2	3	1
3E	1	1	1	1	1	5	2	1	1	1
4E	3	2	4	1	5	1	5	1	3	1
5E	3	2	3	2	3	2	3	2	3	2
6E	1	4	1	3	1	5	1	3	1	1
7E	2	4	3	3	3	3	2	4	3	4
8E	1	3	5	1	4	2	5	2	5	1

	Scales		
SUS	Usability	Learnability	
60,3	53,9	85,9	
87,5	84,4	100,0	
67,5	59,4	100,0	
42,5	28,1	100,0	
85,0	81,3	100,0	
62,5	59,4	75,0	
22,5	9,4	75,0	
37,5	37,5	37,5	
77,5	71,9	100,0	

Age	Tests	Gender
62	70	
44	125	
38	53	
19	7	
22	13	
31	7	
40	101	
40	4	

Task Method

Mean SUS Score	48,1
StDev	28,0

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1T	1	5	1	1	1	5	1	5	1	1
2T	3	1	5	1	4	1	5	1	5	1
3T	1	3	3	2	2	5	3	2	1	2
4T	1	4	1	5	1	5	3	5	1	3
5T	2	2	4	2	3	1	3	2	3	2
6T	1	4	2	2	1	5	2	4	2	3
7T	2	2	4	2	4	3	4	1	2	1
8T	2	2	4	1	3	4	4	2	2	1

	Scales	
SUS	Usability	Learnability
48,1	40,6	78,1
20,0	0,0	100,0
92,5	90,6	100,0
40,0	31,3	75,0
12,5	9,4	25,0
65,0	62,5	75,0
25,0	15,6	62,5
67,5	62,5	87,5
62,5	53,1	100,0

Age	Tests	Gender
31	118	
40	41	
19	6	
52	51	
36	9	
30	36	
19	3	
45	5	

Figure 5.3: The results from the SUS questionnaire, as well as the agem amount of tests and gender of each tester. A bigger version can be found in appendix H.

In section 2.1 the following research question was posed:

Is there a (significant) difference in the testers subjective impression and opinion of the website with each method?

In section 3.8 the following hypotheses were formulated for the purpose of answering the research question by the use of System Usability Scale means:

$$H_0 = \text{There is no significant } (\alpha = 0.05) \text{ difference between the two SUS-score means.} \tag{5.3}$$

and the alternative hypothesis

$$H_1 = \text{There is a significant } (\alpha = 0.05) \text{ difference between the two SUS-score means.} \tag{5.4}$$

An independent two sample t-test was decided upon as suitable to hypothesis test the two means. To find the p-value one needs to first calculate t using the following equation:

$$t = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{5.5}$$

where

$\bar{x_1}$ and $\bar{x_2}$ are the means of sample 1 and sample 2
s_1 and s_2 are the standard deviation of sample 1 and sample 2
n_1 and n_2 are the sample size of sample 1 and sample 2

When the SUS-score results from the tests are inserted we get that:

$$\begin{aligned} t &= \frac{60.3 - 50.3}{\sqrt{\frac{23.8^2}{8} + \frac{28^2}{8}}} \\ t &= \frac{10}{\sqrt{70.8 + 98}} \\ t &= \frac{10}{\sqrt{168.8}} \\ t &= \frac{10}{12.99} \\ t &= 0.77 \end{aligned} \tag{5.6}$$

The degrees of freedom *df* can normally be calculated using the two sample sizes minus 2 as a shortcut:

$$df = n_1 + n_2 - 2 \quad \Rightarrow \quad df = 8 + 8 - 2 = 14 \tag{5.7}$$

However, if the variances are markedly different, then the following formula is used:

$$df' = \frac{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})^2}{(\frac{1}{n_1 - 1})(\frac{s_1^2}{n_1})^2 + (\frac{1}{n_2 - 1})(\frac{s_2^2}{n_2})^2} \tag{5.8}$$

By inserting the data, we get that

$$\begin{aligned} df' &= \frac{(\frac{23.8^2}{8} + \frac{28^2}{8})^2}{(\frac{1}{8-1})(\frac{23.8^2}{8})^2 + (\frac{1}{8-1})(\frac{28^2}{8})^2} \\ df' &= \frac{(70.8 + 98)^2}{(\frac{1}{7})(70.8)^2 + (\frac{1}{7})(98)^2} \\ df' &= \frac{28493.44}{2088.1} \\ df' &= 13.65 \Rightarrow df = 13 \end{aligned} \tag{5.9}$$

As shown, the variances of the two samples are different enough for *df* being calculated differently. We set *df* to be 13. With *t* and *df* calculated we can use the t-table in appendix A to approximate the *p*-value. To get a precise *p*-value the Google Sheets TDIST function can be used by inserting *t*, *df* and the amount of tails being tested for:

$$TDIST(t, df, tails) \quad \Rightarrow \quad TDIST(0.77, 13, 2) \tag{5.10}$$

which equals a *p*-value of

$$p = 0.46 \tag{5.11}$$

A *p*-value of 0.46 tells us that while the exploratory method yielded a 10 point higher SUS-score than the traditional task method, we can only be 54% sure that the SUS-score is significantly different. In other words, there is a 46% probability that the difference is due to chance alone.

We can now hold p against α in the hypothese and see that since $p > \alpha$ the null-hypothesis H_0 can't reliably be rejected, thus H_1 can't be accepted. In other words there is no significant difference between the two methods in regards to the System Usablity Score and **therefore no significant different in the testers subjective impression and opinion of the website with each method**.

5.3 Preferences

The exploratory method group was asked to comment on their opinion of the exploratory approach in relation to the task-based approach, as well as stating their preference. This was done to examine if the testers had a preference to the new method compared to what they are used to. While the exploratory approach might be more cost effective due to the lack of task design, if it is highly unpopular and cumbersome to the testers, it might not be worth it after all. The answers were delivered oral. An overview of the answers can be found in table 5.3 and a transcript of the answers can be found in appendix F.

	1E	**2E**	**3E**	**4E**	**5E**	**6E**	**7E**	**8E**
Preference	Unsure	Task	Unsure	Hybrid	Task	Unsure	Exploratory	Task
λ	0.14	0.28	0.06	0.25	0.33	0.33	0.17	0.17

Table 5.2: The preference of the testers did not seem to have any correlation with the value they brought. The preference of the testers can be found in appendix F.

Curiously enough, the preference of the testers did not seem to have any correlation with the value they brought. The least effective exploratory tester 3E uncovered 2 errors ($\lambda = 0.06$), but was one of the few who had no preference between the two methods:

> *"No difference for me. I think im capable of thinking outside the box and able to independently relate to the site without specific tasks."* - **3E**

Two testers uncovered 12 errors each ($\lambda = 0.33$); 5E and 6E. 5E questioned the amount of value his test had produced and had a clear preference for the task-based approach:

> *"I have just clicked my way around and given my opinion but i don't think that i have given a more concrete and complete opinion or answers to you (UserTribe). To be honest i prefer to be given tasks so i know how to solve this. I'm fine with exploring and having the freedom, but i like having the tasks*

so i know what to look for and attempt to do. That way you can give me and the website a bit more help. So i definitely prefer that you (UserTribe) create the tasks for me. It's hard to have your own opinion."" - **5E**

6E on the other hand didn't have any preference, but she provided some good indsights on the method:

"The exploratory approach seems more real though because it shows what i would have done on my own. I think i reached some conclusions i wouldn't have reached if i had been given a task. Without a task i had to think for myself and relate to the choice i was making and i think the choices were more natural. If i had been asked to go find a specific product and try buying it, eventhough it was a product i wouldn't normally buy it would make the situation unnatural, eventhough i would have tried to relate to the situation. I also probably wouldn't have thought about if the website seemed trustworthy, since it wouldn't have been a part of the task. I don't prefer any over the other. I think it depends on what the test is used for. I think there is more "meat" on the explorative test, but the tester can ofcourse be led astray." - **6E**

As seen in table 5.3 there is a noticable difference in the testers ability to uncover errors. Likewise, noticable differences in the testers ability to immersive themselves in the test was noted while viewing the videos. Some testers just seemed better making use of the exploratory walkthrough method. This leads us to conclude that **there seem to be a difference in the individual testers ability to use the exploratory walkthrough method**. Meanwhile, testers that performed poorly on the test had a perception of the opposite in regards to their own skill. As seen in figure 5.3 neither age, gender or experience seem to have any correlation to how many errors each tester uncovered. Due to this we can answer the last research question by saying that it **does NOT seem to be possible to categorize who better can use the exploratory walkthrough method.**

	Problems	Age	Experience	Gender
3E	2	38	6	
1E	5	62	118	
7E	6	40	3	
8E	6	40	5	
4E	9	19	51	
2E	10	44	41	
5E	12	22	9	
6E	12	31	36	

Figure 5.4: When comparing the age, gender and experience to the amount of problems identified by each tester, there seem to be no apparent correlation.

Chapter 6

Discussion

The main finding from the results was that the exploratory method group mainly uncovered unique errors that were superficial, due to them happening early in the shopping process. This might be cause by the tester not knowing to what extend they are allowed to go with this new method. Some of bad exploratory testers did a classic 'click-and-tell', which was basically them clicking on any category or other link they could find and reading text. No attempt was made to create a personal use case. If the exploratory walkthrough method is to be used, it is my advice that the expectations of the test are cleared out with the tester beforehand. While this was done in the test scenario, it would seem that it is needed in a greater extend.

Another main finding of the study to be discussed was the lack of significant difference in the SUS-scores. The two methods scored 50.3 and 60.3 which is in the 13-29% percentile rank for Raw SUS scores. A difference in means by 10 seems like a lot, but when the p-value was calculated, we could only be 54% that the means were significally different. This however was due to the unusually high standard deviation of the two test; 23.8% and 28%. What has caused the high standard deviation might be due to the decision made in chapter 4 to find a website that is not 'good'. That decision was made due to a concern that if a website has few errors and a high SUS-score the distinction (or lack of) between the two groups will be hard to measure and prove. The decision turned out to result in a problem in the other end of the spectrum.

Chapter 7

Conclusion

The aim of this comparative study was to explore the differences between two walkthrough methods while performing remote asynchronous usability testing. The study found no significant ($p = 0.46$) difference in the subjective opinion of the two groups towards the website.

It was however found that, while there was no significant ($p = 0.786$) difference in the amount of errors each method identified, they discovered errors at different parts of the shopping process. The exploratory method identified all of its unique errors during the first part; browsing and shopping, while the task method identified most of its unique errors during the last part; buying and check-out. This is assumed to be due to the task method group being forced through to the check-out process, while many in the exploratory method group didn't. To counter this, it is advised to give the testers an introduction to the limits (or lack hereof) of the test method and also emphasise that the free rein is given with the expectation of responsibility.

The study found big differences in the performance of the exploratory testers. While the least succesful tester discovered a mere 2 errors, the two best testers uncovered 12 errors each and both went through the check-out part of the shopping process. It was however not possible to find any relation in terms of age, gender or experience.

While the exploratory walkthrough method has a lot of potential, the task based approach still seems to be the best method by showing to have a higher average problem detection rating $\lambda = 0.3$ versus the lower $\lambda = 0.22$. The detection rating of the exploratory walkthrough method might improve when a sufficient way to introduce the testers to the method has been found. While it is my assumption that every tester *can* learn to use the method, an obvious next step in research could be to explore if that is the case.

Bibliography

[Anders and Simon] Anders, K. and Simon, H. A. Verbal reports as data. 87(3):215–251. Cited on 9

[Bangor et al.] Bangor, A., Kortum, P. T., and Miller, J. T. An empirical evaluation of the system usability scale. 24(6):574–594. Cited on 10

[Bonde] Bonde, K. En krone for dine tanker - en komparativ analyse af tænke-højt metoden. Cited on 10

[Boren and Ramey] Boren, T. and Ramey, J. Thinking aloud: reconciling theory and practice. 43(3):261–278. Cited on 9

[Brooke] Brooke, J. SUS-a quick and dirty usability scale. 189(194):4–7. Cited on 10

[Bruun and Stage] Bruun, A. and Stage, J. The effect of task assignments and instruction types on remote asynchronous usability testing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2117–2126. ACM. Cited on 8, 9

[Castillo et al.] Castillo, J. C., Hartson, H. R., and Hix, D. Remote usability evaluation: Can users report their own critical incidents? In *CHI 98 Cconference Summary on Human Factors in Computing Systems*, CHI '98, pages 253–254. ACM. Cited on 8

[Cordes] Cordes, R. E. Task-selection bias: A case for user-defined tasks. 13(4):411–419. Cited on 8

[Dray and Siegel] Dray, S. and Siegel, D. Remote possibilities?: International usability testing at a distance. 11(2):10–17. Cited on 7

[Hartson and Castillo] Hartson, H. R. and Castillo, J. C. Remote evaluation for post-deployment usability improvement. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '98, pages 22–29. ACM. Cited on 9

[Hartson et al.] Hartson, H. R., Castillo, J. C., Kelso, J., and Neale, W. C. Remote evaluation: The network as an extension of the usability laboratory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 228–235. ACM. Cited on 7, 9

[Hix et al.] Hix, D., Rosson, M. B., Williges, R. C., Castillo, J. C., and Castillo, J. C. The user-reported critical incident method for remote usability evaluation. Cited on 8, 9

[Holmes and Joyce] Holmes, B. J. and Joyce, D. T. *Object-oriented Programming with Java*. Jones & Bartlett Learning. Google-Books-ID: 271wpK2CQ0EC. Cited on 19

[Jeffries et al.] Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. User interface evaluation in the real world: A comparison of four techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 119–124. ACM. Cited on 7

[Millen] Millen, D. R. Remote usability evaluation: User participation in the design of a web-based email service. 20(1):40–45. Cited on 9

[16] Nielsen, J. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 373–380. ACM. Cited on 6, 7

[17] Nielsen, J. *Usability Engineering*. Morgan Kaufmann Publishers Inc. Cited on 19

[18] Nielsen, J. Usability inspection methods. pages 25–62. John Wiley & Sons, Inc. Cited on 6

[Nielsen and Landauer] Nielsen, J. and Landauer, T. K. A mathematical model of the finding of usability problems. pages 206–213. ACM Press. Cited on 12

[Nielsen and Molich] Nielsen, J. and Molich, R. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 249–256. ACM. Cited on 5

[Rubin] Rubin, J. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Inc., 1st edition. Cited on 7

[Sauro] Sauro, J. *A practical guide to the System Usability Scale: Background, benchmarks, & best practices*. Cited on 10

[Sauro and Lewis] Sauro, J. and Lewis, J. R. *Quantifying the User Experience: Practical Statistics for User Research*. Morgan Kaufmann. Google-Books-ID: USPfCQAAQBAJ. Cited on 11

[24] Scholtz, J. Adaptation of traditional usability testing methods for remote testing. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001*, pages 8 pp.–. Cited on 9

[25] Scholtz, J. A case study: Developing a remote, rapid, and automated usability testing methodology for on-line books. In *Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 2 - Volume 2*, HICSS '99, pages 2015–. IEEE Computer Society. Cited on 9

[Scholtz and Downey] Scholtz, J. and Downey, L. Methods for identifying usability problems with web sites. In Chatty, S. and Dewan, P., editors, *Engineering for Human-Computer Interaction*, number 22 in IFIP —The International Federation for Information Processing, pages 191–206. Springer US. DOI: 10.1007/978-0-387-35349-4_11. Cited on 9

[Thompson] Thompson, J. A. Investigating the effectiveness of applying the critical incident technique to remote usability evaluation. Cited on 8

[Tullis and Stetson] Tullis, T. S. and Stetson, J. N. *A Comparison of Questionnaires for Assessing Website Usability*. Cited on 10, 11

[Virzi] Virzi, R. A. Refining the test phase of usability evaluation: How many subjects is enough? 34(4):457–468. Cited on 12

Appendix A

Appendix

t Table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	**0.50**	**0.25**	**0.20**	**0.15**	**0.10**	**0.05**	**0.025**	**0.01**	**0.005**	**0.001**	**0.0005**
two-tails	**1.00**	**0.50**	**0.40**	**0.30**	**0.20**	**0.10**	**0.05**	**0.02**	**0.01**	**0.002**	**0.001**
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

Appendix B

Testscenario:

Du står og mangler at købe en fødselsdagsgave til en ven. Du har hørt at hjemmesiden www.trendway.dk har et stort udvalg af forskellige varer og vælger derfor at besøge den.

Under løsningen af de kommende opgaver bedes du tænke højt, og sige hvad der undrer dig, samt hvad der er godt og skidt.

Gå nu til: www.trendway.dk

Testopgaver:

1. Hvad er dit første indtryk af siden? Tænk højt, og beskriv dine første indskydelser. (Brug 2 min.)

2. Find nu en eller flere fødselsdagsgaver til din ven og tilføj den/dem til kurven. Du har besluttet dig for et rådighedsbeløb på 800kr. (Brug 7 min.)

3. Find nu en varer du personligt syntes om og tilføj den til din kurv. Beskriv din oplevelse undervejs (Brug 3 min.)

4. Du vil nu gennemføre købet af dine varer. Find selv på personlige oplysninger, indtast en mail der slutter med @usertribe.com og lav en tilfældig kode). Kig på hvert step, mens du tænker højt. Stop ved indtastning af kortoplysninger. (Brug 3 min.)

5. Oplever du problemer under gennemførslen af købet og indtastningen af de personlige oplysninger? Begrund dit svar. (Brug 2 min.) A. Hvis ja, er det så nemt at finde en måde at kontakte kundeservice?

6. Du vil gerne vide hvor lang tid reklamationsretten er hos Trendway.dk. Hvad gør du? Beskriv din oplevelse undervejs. (Brug 2 min.)

7. Hvad er dit overordnede indtryk af Trendway.dk? Hvad fungerer godt og hvad fungerer mindre godt? (Brug 2 min.)

8. Ville du personligt føle dig tryg ved at handle online ved Trendway.dk? Hvorfor/ hvorfor ikke? (Brug 1 min.)

Appendix C

Testscenario:

You have to buy a birthday gift for a friend. You've heard that thewebsite www.trendway.dk has a big selection of various products and thereforechooses to visit it

During the completion of the following tasks you are asked to think aloud, to tell what puzzles you and what is good or bad.

Go now to: www.trendway.dk

Test Tasks:

1. What is your first impression of the website? Think aloud and describe your first thoughts. (Spend 2 min.)
2. Find one or more birthday gifts for your friend and add it/themto the basket. You have decided to spend a maximum of 800k. (Spend 7 min.)
3. Find a product you personally like and add it to the basket. Describe your experience along the way. (Spend 3 min.)
4. You now want to make a purchase of the chosen products. Make up personal information and use an e-mail ending in @usertribe.com and create a random code. Look at each step while thinking aloud. Stop when having to enter creditcard details. (Spend 3 min.)
5. Do you experience any problems while completing your purchase and when entering your personal information? Justify your answer. (Spend 2 min.) A. If yes, is it easy to find a way to contact customer service?
6. You want to know the time on the right of cancellation at Trendway.dk. What do you do? Describe your experience along the way. (Spend 2 min.)

7. What is your overall impression of Trendway.dk. What works well and what doesn't work quite as well (Spend 2 min.)

8. Would you personally feel safe shopping at Trendway.dk? (Spend 1 min.)

Appendix D

Testscenario:

Dette er en eksplorativ test, hvilket betyder der vil være få eller ingen testopgaver. Testen varer stadig 20-30 minutter, men du vil have friere tøjler til at undersøge hvad du finder interessant på siden.

Husk som altid at tænke højt! :)

Testopgaver:

1. Begynd først med at beskrive hvad der er vigtigt for dig når du shopper online og hvilke præferencer/behov du har. Er du tryg ved det? Spørger du nogen til råds? Sammenligner du priser? (Brug 5 min)

2. Gå nu ind på www.trendway.dk. Gå på opdagelse på siden og tag udgangspunkt i dine egne interesser. Tag dig god tid og vær grundig. Husk at tænke højt og beskrive dine indskydelser. (Brug 16 min)

3. Efter at have udført testen svar da på følgende spørgsmål:

   Hvad er din mening om den tilgang til at teste, du netop har prøvet (eksplorativ), i forhold til den metode der normalt bliver anvendt hos UserTribe (testopgaver)? (Brug 2 min)

   Hvilken metode foretrækker du? Hvorfor? (Brug 3 min)

Appendix E

Testscenario:

This is an exploratory test, which means that there will be few or no test tasks. The test still requires 20-30 minutes to complete, but you will have freer reins to explore what you find interesting on the page.

Remember as always to think-aloud! :)

Test Tasks:

1. Start first by describing what is important to you when you shop online and what preferences/needs you have. Do you feel comfortable with it? Do you ask anyone for advice? Do you compare prices? (Spend 5 min)

2. Go now to www.trendway.dk. Start exploring the website and go by your own interests. Take your time and be thorough. Remember to think-aloud and describe what is on your mind. (Spend 16 min)

3. After completing the test answer the following questions:

   What is your opinion on the method of testing you have just tried (exploratory) in comparison to the method normally used at UserTribe (tasks)? (Spend 2 min)

   What method do you prefer? Why? (Brug 3 min)

Appendix F

Tester ID: **1E**

Depends on the website being tested. Personally thinks hes good because he likes talking and is curious. In comparison to what is usually being done (task). It can be hard to stay within the time limit of the tasks. Usually there is too little time to do the tasks, atleast if you know how to express yourself and have something to say. What method do you prefer? Its hard to answer. Everything at its own time. Its nice to be able to talk freely from your heart without the time constraints. On the other hand, if youre unfamiliar with the website being tested and youre unsure, its nice having some tasks as a support. I don't know what I prefer. It depends on what is being tested.

Tester ID: **2E**

I actually prefer task-based. I like getting a concrete task. Because if there isn't a concrete task, i'll just be searching aimlessly. Like if it is a site that doesn't have anything i really need. If you can't come up with any purpose of visiting the site, if there is anything i need. I prefer it when there is a task saying 'You need to find a scale for your mothers birthday' - fine! Also then you can hear - i became very negative towards this website and i've never tried that before. And that has something to do with - what is my purpose here? It doesn't appeal to me. If my task had been "find a scale" or "try buying a product" or "find a wine cabinet" or "find a gift for 500kr" then i wouldn't have ended up being so negative towards the website. Because then i had only had to relate to the tasks i had to solve or if the question had been "whats your first impression of the website" then i would had said that i think it looks simple and so on. But yes, its two very different ways of testing. It might be

good for some uses, but thats ofcourse something you (UserTribe) know, when it's good. But i honestly prefer to get a concrete task. It makes it easier. And it means that it's completely different subject i talk about.

Tester ID: **3E**

I assume that this was a fictive website for testing purposes. I think the method is fine. There are both pros and cons. If one can't limit it himself then he might drift off too much, but ofcourse sometimes, if the tasks are too leading, on how you are supposed to experience a website then you'll already have an imposed opinion. So i'm fine with it. What method do i prefer? No difference for me. I think im capable of thinking outside the box and able to independently relate to the site without specific tasks. Though on the other side, it's nice to have some specific tasks to have to relate to when you need to find a product, to see how long it takes. I can't exclude one or the other. I like them both.

Tester ID: **4E**

I think it is pretty cool. I have an interest in websites and i think it's both exciting to get some boundaries to talk within, but also just being allowed to talk freely. On this website there were more things to talk about; the lack of a dropdown menu, that's what i mostly wanted to talk about. There isn't much other than the green color. No pictures other than the landingpage.I thought it was interesting to be allowed to choose what to talk about. But i don't know. What do i prefer? Then.. I think i lean more towards having a theme on what to talk about. Being bound on time might be good for some, but you shouldn't have to keep talking if there is no more to talk about. So that i don't like. I think it's fine that there is a guide on time for other users, but what i prefer is that there is a theme to guide you. I want to know what you (UserTribe) want to know about the website. Do you want to know something about the layout, the colors, the pictures or the menu? It would be nice if that was included in the description. Then you'll get that information instead of me just rambling about stuff that has no interest to you. So i would say it would be nice if there were some few tasks/questions that also requires the tester to elaborate. So, summary: few basic questions, don't be bound on too

much time (or atleast don't be too strict), ask the testers to "elaborate" on their answers or give them an opportunity to talk freely outside of the tasks, the testers first and last thoughts on the website.

Tester ID: **5E**

It's fine that i can explore the website on my own, but it would be nice to have some tasks so i know how to deal with the website. So i know how to search for the things. It's fine that i can just explore, but i don't feel that i can give better answers and opinions if i don't have some tasks. I have just clicked my way around and given my opinion but i don't think that i have given a more concrete and complete opinion or answers to you (UserTribe). To be honest i prefer to be given tasks so i know how to solve this. I'm fine with exploring and having the freedom, but i like having the tasks so i know what to look for and attempt to do. That way you can give me and the website a bit more help. So i definitely prefer that you (UserTribe) create the tasks for me. It's hard to have your own opinion. Again, i prefer you creating the tasks for me. It's fine that you try out this new method, but you shouldn't spend too much time on it. Maybe 1 task could be "Spend 5 minutes exploring the website" and then follow it up with tasks.

Tester ID: **6E**

In my opinion it is easiest to test using tasks instead of exploratory since you know what is expected of you and you therefore avoid breaks. The exploratory approach seems more real though because it shows what i would have done on my own. I think i reached some conclusions i wouldn't have reached if i had been given a task. Without a task i had to think for myself and relate to the choice i was making and i think the choices were more natural. If i had been asked to go find a specific product and try buying it, eventhough it was a product i wouldn't normally buy it would make the situation unnatural, even-though i would have tried to relate to the situation. I also probably wouldn't have thought about if the website seemed trustworthy, since it wouldn't have been a part of the task. I don't prefer any over the other. I think it depends on what the test is used for. I think there is more "meat" on the explorative

test, but the tester can ofcourse be led astray. Basically I think the task based method is easier because you follow a "recipe" and the exploratory method is a bit more challenging, though not necessarily in a bad way, because it is also interesting to get to explore the website on your own and it definitely let's you think more.

Tester ID: **7E**

I think it functions well because you get free rein to familiarize yourself with the website in another way. It let's me explore the website more and when you explore the website you think-aloud and say what you think. So i think you shouldn't answer so direct. On the other hand then the time might be spent searching for something that doesn't capture ones interest, so it's a benefit to have something to work towards. But it's a new, positive and interesting way to explore the website and then see what captures the customers when they get thrown in on their own. What is it they notice? Can one find errors that way? Something that could be better? Or things that irks you? That's things you would think-aloud and say that way. What method do i prefer? Hm that's a tough question because i don't think you can compare those two things. I think i would prefer number one (task), but i'm not sure because it's probably beneficial to just explore and think-aloud and come with some thoughts that are different than normal. Since it's the first time i think i would prefer the old method, but i'm actually in a lot of doubt. I might actually say i prefer this one (exploratory) since, i've only tried it once. It's a new way of thinking and if it gives a bigger insight in how the website is interpreted by the users, if it's different with this method, then it is probably better.

Tester ID: **8E**

I think it's interesting testing this way, you ofcourse need to go through a lot of things and need to think about what one is doing the situation you are in. One ofcourse needs to talk yourself through many things and note if you're using enough time to go through everything. I think it's harder to use this method. Which one do i prefer? I'd say that i prefer the normal method. You are being guided more. The explorative way requires you to make your own tasks and think about what you need. So yes, i prefer the normal way.

Appendix G

Deg	Name	1T	2T	3T	4T	5T	6T	7T	8T	1E	2E	3E	4E	5E	6E	7E	8E	Freq (λ)	Name
1	P1	x		x	x	x	x	x	x	x		x	x	x	x	x	x	0.88	P1
2	P12		x	x	x	x	x	x			x		x	x	x			0.63	P12
3	P20			x	x		x	x					x	x	x	x	x	0.56	P20
2	P11	x		x	x		x				x			x	x		x	0.50	P11
3	P15	x		x	x	x	x	x	x		x							0.50	P15
1	P16	x				x	x	x				x			x	x	x	0.50	P16
2	P4								x	x			x	x		x	x	0.38	P4
3	P9					x	x				x		x		x	x		0.38	P9
3	P10			x	x		x	x	x		x							0.38	P10
3	P13	x		x	x		x				x				x			0.38	P13
1	P6	x		x		x					x			x				0.31	P6
1	P7	x				x		x			x			x				0.31	P7
1	P22						x	x						x	x		x	0.31	P22
2	P30		x			x	x	x							x			0.31	P30
3	P35			x	x	x	x		x									0.31	P35
3	P27	x		x		x			x									0.25	P27
3	P29	x		x	x	x												0.25	P29
2	P8			x							x					x		0.19	P8
1	P17						x	x					x					0.19	P17
3	P28	x	x						x									0.19	P28
3	P31		x						x						x			0.19	P31
2	P34			x		x		x										0.19	P34
3	P18					x							x					0.13	P18
1	P19						x						x					0.13	P19
1	P23													x	x			0.13	P23
1	P25													x	x			0.13	P25
3	P33		x				x											0.13	P33
1	P2									x								0.06	P2
2	P3									x								0.06	P3
1	P5									x								0.06	P5
3	P14										x							0.06	P14
1	P21												x					0.06	P21
3	P24													x				0.06	P24
2	P26													x				0.06	P26
1	P32		x															0.06	P32
2	P36						x											0.06	P36
	#	10	6	13	9	13	16	11	8	5	10	2	9	12	12	6	6	0.2569	
	λ	0.28	0.17	0.36	0.25	0.36	0.44	0.31	0.22	0.14	0.28	0.06	0.25	0.33	0.33	0.17	0.17		
		Task λ avg: 0.30								Exploratory λ avg: 0.22									

Appendix H

Exploratory Method

Mean SUS Score	60,3
StDev	23,8

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1E	1	2	5	1	5	1	5	1	5	1
2E	1	1	4	1	4	4	4	2	3	1
3E	1	1	1	1	1	5	2	1	1	1
4E	3	2	4	1	5	1	5	1	3	1
5E	3	2	3	2	3	2	3	2	3	2
6E	1	4	1	3	1	5	1	3	1	1
7E	2	4	3	3	3	3	2	4	3	4
8E	1	3	5	1	4	2	5	2	5	1

	Scales		
	SUS	Usability	Learnability
	60,3	53,9	85,9
1E	87,5	84,4	100,0
2E	67,5	59,4	100,0
3E	42,5	28,1	100,0
4E	85,0	81,3	100,0
5E	62,5	59,4	75,0
6E	22,5	9,4	75,0
7E	37,5	37,5	37,5
8E	77,5	71,9	100,0

	Age	Tests	Gender
1E	62	70	
2E	44	125	
3E	38	53	
4E	19	7	
5E	22	13	
6E	31	7	
7E	40	101	
8E	40	4	

Task Method

Mean SUS Score	48,1
StDev	28,0

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1T	1	5	1	1	1	5	1	5	1	1
2T	3	1	5	1	4	1	5	1	5	1
3T	1	3	3	2	2	5	3	2	1	2
4T	1	4	1	5	1	5	3	5	1	3
5T	2	2	4	2	3	1	3	2	3	2
6T	1	4	2	2	1	5	2	4	2	3
7T	2	2	4	2	4	3	4	1	2	1
8T	2	2	4	1	3	4	4	2	2	1

	Scales		
	SUS	Usability	Learnability
	48,1	40,6	78,1
1T	20,0	0,0	100,0
2T	92,5	90,6	100,0
3T	40,0	31,3	75,0
4T	12,5	9,4	25,0
5T	65,0	62,5	75,0
6T	25,0	15,6	62,5
7T	67,5	62,5	87,5
8T	62,5	53,1	100,0

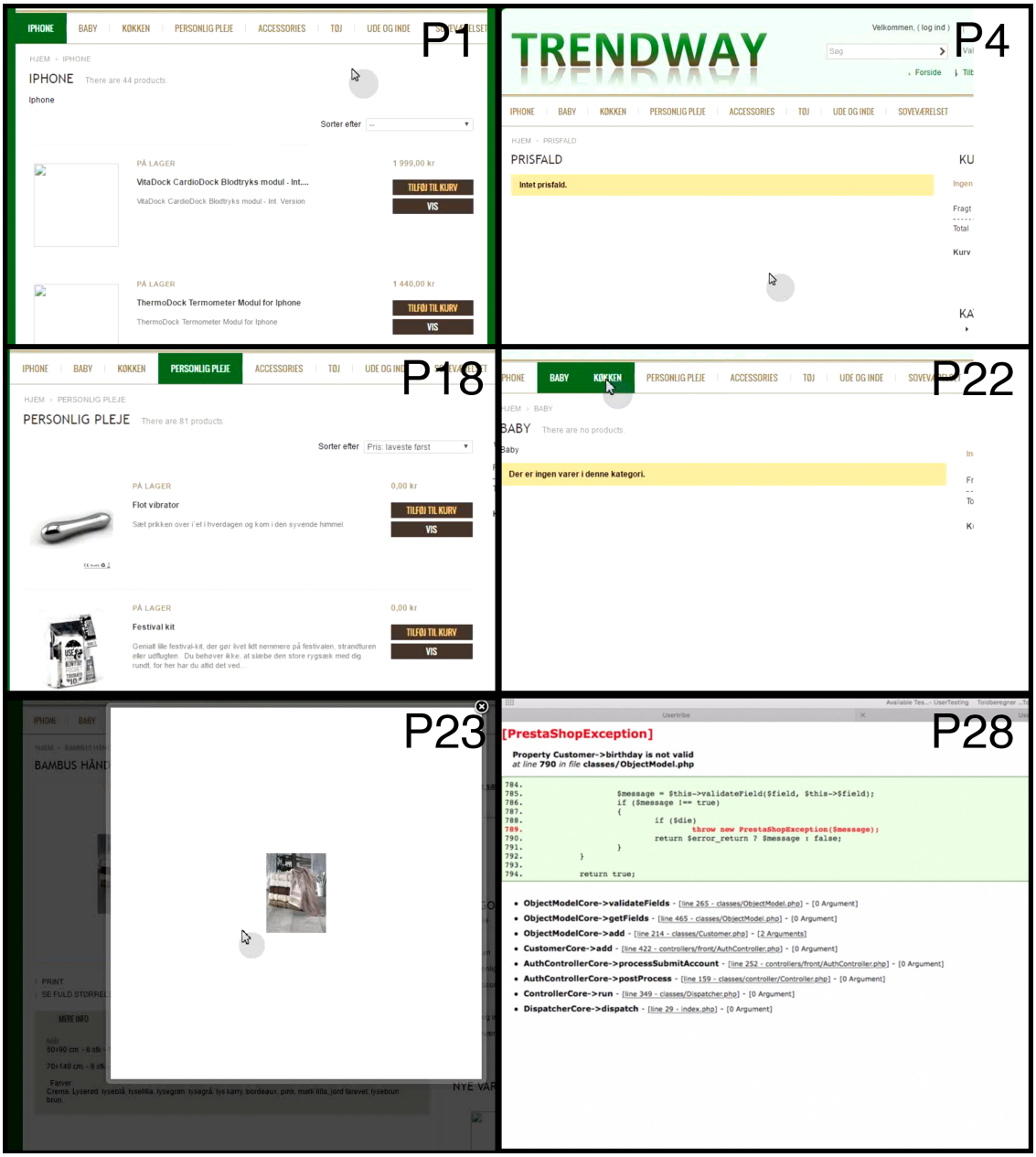
	Age	Tests	Gender
1T	31	118	
2T	40	41	
3T	19	6	
4T	52	51	
5T	36	9	
6T	30	36	
7T	19	3	
8T	45	5	

Appendix I

	Problem ID	Description
1	**P1**	Pictures are not loading
1	**P2**	The user assumed the clothes category only contains womens clothing (Not enough overview)
2	**P3**	"About us" doesn't contain sufficient information
2	**P4**	"Tilbud" doesn't contain any information
1	**P5**	Same products are present on different pages
1	**P6**	It is not clear that the pictures on the frontpage are categories
1	**P7**	Two categories have the same picture
2	**P8**	"Most popular products" doesn't show anything
3	**P9**	No subcategories
3	**P10**	Can't choose size/color on product when adding to basket
2	**P11**	Need more information on product
2	**P12**	Can't add product to basket / Only shows after reload
3	**P13**	Overview on payment and handling is confusing/doesn't make sense
3	**P14**	No information on handling costs in terms
3	**P15**	Need to enter adress twice in check out flow
1	**P16**	Products are placed in the wrong categories
1	**P17**	Pixelated picture on frontpage
3	**P18**	Product costing 0kr

	Problem ID	Description
1	**P19**	No clear phone number in the "Contact" category
3	**P20**	"Om os" and "Kunderoplysninger m.m" etc. aren't working
1	**P21**	"Read more" under "Nye varer" hasn't been translated
1	**P22**	No products in the "Baby" category
1	**P23**	Small product picture / Need ability to zoom
3	**P24**	Changing the amount of items shown per page gives an error
1	**P25**	No adress under "Contact us"
2	**P26**	Clicking "Min konto" leads to the check out page
3	**P27**	Dead link when clicking big picture on front page
3	**P28**	The error message is not understandable by the user
3	**P29**	Trying to correct an error during the checkout process by pressing the button "back" sends the user to the frontpage
2	**P30**	Missing sort option
3	**P31**	Search function doesn't allow misspelling of a word, needs improvement
1	**P32**	Missing gallery view of products
3	**P33**	Basket dissapears
2	**P34**	Missing info box for clarification
3	**P35**	No carriers deliver to real address
2	**P36**	Basket doesn't update when products are removed

Appendix J


P1
IPHONE
BABY
KØKKEN
PERSONLIG PLEJE
ACCESSORIES
TØJ
UDE OG INDE
IPHONE
There are 44 products.
Iphone
Sorter efter
PÅ LAGER
1 999,00 kr
VitaDock CardioDock Blodtryks modul - Int....
TILFØJ TIL KURV
VIS
PÅ LAGER
1 440,00 kr
ThermoDock Termometer Modul for Iphone
TILFØJ TIL KURV
VIS
P4
TRENDWAY
Velkommen, (log ind)
Søg
Forside
IPHONE
BABY
KØKKEN
PERSONLIG PLEJE
ACCESSORIES
TØJ
UDE OG INDE
SOVEVÆRELSET
HJEM » PRISFALD
PRISFALD
Intet prisfald.
Ingen
Fragt
Total
Kurv
P18
IPHONE
BABY
KØKKEN
PERSONLIG PLEJE
ACCESSORIES
TØJ
HJEM » PERSONLIG PLEJE
PERSONLIG PLEJE
There are 81 products.
Sorter efter
Pris: laveste først
PÅ LAGER
0,00 kr
Flot vibrator
Sæt prikken over i'et i hverdagen og kom i den syvende himmel.
TILFØJ TIL KURV
VIS
PÅ LAGER
0,00 kr
Festival kit
Genialt lille festival-kit, der gør livet lidt nemmere på festivalen, strandturen eller udflugten. Du behøver ikke, at slæbe den store rygsæk med dig rundt, for her har du altid det ved...
TILFØJ TIL KURV
VIS
P22
IPHONE
BABY
KØKKEN
PERSONLIG PLEJE
ACCESSORIES
TØJ
UDE OG INDE
HJEM » BABY
BABY
There are no products.
Baby
Der er ingen varer i denne kategori.
P23
BAMBUS HÅNDKLÆDE
PRINT
SE FULD STØRRELSE
MERE INFO
NYE VARER
P28
[PrestaShopException]
Property Customer->birthday is not valid
at line 790 in file classes/ObjectModel.php
784.
785. $message = $this->validateField($field, $this->$field);
786. if ($message !== true)
787. {
788. if ($die)
789. throw new PrestaShopException($message);
790. return $error_return ? $message : false;
791. }
792. }
793.
794. return true;
ObjectModelCore->validateFields - [line 265 - classes/ObjectModel.php] - [0 Argument]
ObjectModelCore->getFields - [line 465 - classes/ObjectModel.php] - [0 Argument]
ObjectModelCore->add - [line 214 - classes/Customer.php] - [2 Arguments]
CustomerCore->add - [line 422 - controllers/front/AuthController.php] - [0 Argument]
AuthControllerCore->processSubmitAccount - [line 252 - controllers/front/AuthController.php] - [0 Argument]
AuthControllerCore->postProcess - [line 159 - classes/controller/Controller.php] - [0 Argument]
ControllerCore->run - [line 349 - classes/Dispatcher.php] - [0 Argument]
DispatcherCore->dispatch - [line 29 - index.php] - [0 Argument]