Master thesis

Predicting persistency of usability problems based on error classification

A longitudinal study on improving mobility for the elderly

Ruud Zandbergen

April 2015

Faculty of Behavioural, Management and Social sciences (BMS)

University of Twente

Enschede, the Netherlands

UNIVERSITY OF TWENTE.

Table of contents

Preface

This study was performed as a master thesis for psychology at the University of Twente. The data collection took place during an internship at the National Fund for the Elderly (NFE). I want to thank a number of people who have helped me greatly in finishing this master thesis.

First of all I would like to thank my supervisors Martin Schmettow, Matthijs Noordzij (University of Twente) and Nina van der Vaart (NFE) for their support, expertise, feedback and fresh thoughts about the thesis and project.

I would also like to thank Deborah Oosting for helping me throughout the course of the thesis. It was very valuable to exchange ideas and discuss with you about our theses.

I would like to thank everyone at the NFE and the consortium of the MOBILE.OLD project for helping me to obtain hands-on experience in a very pleasant work environment and treating me like an equal in the discussions and project meetings. Going to London and partaking in one of the project meetings was one of the highlights of the internship.

A big thanks to the participants of this study and to the NFE and the Koperhorst for finding the participants and providing a testing location.

Finally, I would like to thank my girlfriend, friends and family for their support during the entire process. I could not have done it without you.

Abstract

English

Social isolation and loneliness are becoming increasingly serious problems among the elderly. MOBILE.OLD is a project which helps elderly stay independent, healthy and mobile by creating services for mobile devices. This study evaluated the usability of the designed prototype services for this project. Earlier studies have shown the need to look at persistency of usability problems for elderly users when you want to get a clear image of how well your new product is learned by them, as the elderly need a little bit more time to 'get started'. For this reason longitudinal study designs are very appropriate for elderly users. However, due to deadlines and budget restraints, longitudinal designs are often not used. This study wanted to use the results from the MOBILE.OLD project to predict persistency of problems. If this would be possible, predictive measures could become a cheaper alternative to the longitudinal design. Prediction of persistency was attempted by using error classifications for the usability problems. An extended matching protocol was created to incorporate the error classifications in the matching steps. To help evaluators classify the incidents to error categories, a step-by-step classification guideline was constructed. Previous experience and technology enthusiasm or 'geekism' were used as predictors for persistency. A sample of twenty elderly users between the age of 61 and 82 tested ten different applications. Data was collected by capturing video, questionnaires and think aloud procedures. The longitudinal data was used to create persistency patterns for the problems. Three different groups of persistency patterns were further investigated: disappear early, appear late and persistent. Elderly encountered mostly knowledge-based problems and geekism was shown to influence the number of KBFR problems that users encountered. It proved to be difficult to predict the persistency of the problems that elderly encountered during their learning efforts, but some findings were promising in supplying further inspiration for new studies on persistency.

Nederlands

Sociaal isolement en eenzaamheid is een groeiend probleem onder ouderen. MOBILE.OLD is een project dat ouderen helpt om zelfstandig, gezond en mobiel te blijven, door de ouderen te ondersteunen met diensten voor mobiele apparaten. Deze studie heeft de gebruiksvriendelijkheid van de ontwikkelde prototype diensten geëvalueerd voor dit project. Eerdere studies hebben aangetoond dat het heel belangrijk is om bij ouderen rekening te houden met de hardnekkigheid van gebruiksvriendelijkheidsproblemen om een goed beeld te krijgen van hoe ouderenleren omgaan met het nieuwe product. Ouderen hebben namelijk een beetje extra tijd nodig om 'op te starten'. Dit betekent dat een longitudinale onderzoeksopzet erg geschikt is voor het testen van ouderen. Door deadlines en budgettaire restricties wordt vaak voor een andere opzet gekozen. Deze studie heeft geprobeerd om de resultaten van het MOBILE.OLD project te gebruiken om probleemhardnekkigheid te voorspellen. Dit zou een goedkoper alternatief kunnen bieden voor de longitudinale onderzoeksopzet. Het voorspellen van hardnekkigheid is gedaan door foutclassificaties te gebruiken voor de gebruiksvriendelijkheidsproblemen. Een uitgebreid 'matching' protocol is gebruikt om de foutclassificaties vast te stellen voor de problemen. Om evaluatoren te helpen met incidenten in foutclassificaties in te delen, is een stapsgewijze classificatiehandleiding geschreven. Eerdere ervaring en enthousiasme voor techniek of 'geekism' werden gebruikt als voorspellers voor hardnekkigheid. Een steekproef van twintig ouderen tussen de leeftijd van 61 en 82 testte tien verschillende diensten. De data werd verzameld door middel van video opnames, vragenlijsten en 'think aloud' procedures. De longitudinale data is gebruik om patronen voor hardnekkigheid op te stellen voor de problemen. Drie verschillende groepen van patronen zijn verder onderzocht: verdwijnt vroeg, verschijnt laat en hardnekkig. Ouderen bleken voornamelijk kennis-gebaseerde problemen te vinden en geekism bleek het aantal KBFR problemen te beïnvloeden die ouderen vonden. Het bleek lastig om hardnekkigheid te voorspellen voor de problemen van ouderen, maar de resultaten inspireerden wel nieuwe mogelijke onderzoeken naar probleemhardnekkigheid.

Introduction

Social isolation and loneliness are becoming increasingly serious problems among the elderly. With events like getting a retirement from work, loss of a partner, family member or friend and a decrease in mobility, loneliness is lurking for the elderly of 65 years and over (Centraal Bureau voor de Statistiek [CBS], 2012). Social isolation is also negatively influencing the psychological state of the elderly (Tomaka, Thompson & Palacios, 2006), with a higher percentage of the elderly showing depressive symptoms due to social isolation than often is thought (van't Veer-Tazelaar et al., 2008). With a population that is ageing more every year, it is very important to address the problems which elderly are facing and to keep them socially engaged. New and innovative ways to help elderly stay mobile and socially engaged are always sought after. In this day and age the possibilities to help elderly with socially oriented activities have become wider than they have ever been as wireless internet is available almost everywhere for mobile devices. Bargh & McKenna (2004) called the internet the latest in a series of technological breakthroughs in interpersonal communication, following the telegraph, telephone, radio and television, but mobile internet seems to be another big step further in technological advancement. The new possibilities have led to a number of projects focussed on helping elderly to become engaged more in society and to become more mobile by using mobile applications to prevent them from becoming lonely and isolated. The European Union has started to subsidise a number of projects which try to solve problems that elderly face by utilizing innovative technologies. One of these projects is the MOBILE.OLD project, which is a collaboration of companies throughout Europe that focusses on supporting elderly to stay independent, healthy and mobile. This is accomplished by introducing mobile devices with highly specialized residential and outdoor services (Ambient assisted living joint programme [AAL], 2012). The National Fund for the Elderly (NFE) from the Netherlands, which is one of the end-user testing partners in the MOBILE.OLD consortium, commissioned a study to evaluate the usability of the designed prototype services. Besides the usability evaluation, the way elderly learned to use the prototypes and what kind of problems they faced in the process was a big focus point of this study.

Developing for the elderly

It may seem like a plausible possibility for the MOBILE.OLD project to use existing applications to support the elderly, as an enormous variety of applications are already available. To show just how large this market is: 46 billion apps were downloaded in the year 2012 and that number was expected to double in 2013 (Portio Research Limited, 2013). Even though there are so many existing applications, using them for the MOBILE.OLD project proved to be rather difficult, because the elderly are a very different kind of user group than the ‘average’ computer users (Hawthorn, 2003; Shneiderman, 2000). Existing applications are therefore not always appropriate for them to use. Elderly users are often overlooked by design companies when new services and applications are introduced (Rice & Alm, 2008), as not many elderly are expected to use the applications anyway. The elderly are often reluctant to use new technical devices as they are uncertain about how to get started with new devices and think of them as too complicated (Eastin and LaRose, 2006). To help the elderly overcome these fears, the project chose to develop new, user-friendly applications that would be delivered in a highly personalised and intuitive way for elderly. According to Hawthorn (2003) it is not enough to look at guidelines that were made for elderly to accomplish this, but you also need to involve the elderly in the testing and really need to listen to their wishes for improving the applications. To achieve this in the MOBILE.OLD project, the elderly were personally involved in the design and redesign of the applications during the various testing phases. User interviews, card sorting tests and mock-up tests were performed with elderly users during earlier stages of the MOBILE.OLD project, leading to prototype versions of the mobile services that incorporated the earlier received feedback of the elderly. These prototypes were capable of executing almost all basic functions and some advanced functions that were designed. This study took place during the first prototype testing phase, where usability testing was performed for the project to investigate how the elderly users would interact with the services and which parts of the services would be problematic to them. These problems could then be used as input for the next design phase.

Problem severity and persistency

Due to deadlines and budget restraints almost every design project has a limited time span and limited resources for each design phase. Developers try to use these resources to solve as many problems in usability as possible. These usability problems can be defined as issues that influence the effectiveness, efficiency and/or satisfaction of using a system (ISO, 2008; Hornbæk, 2006). As it is often impossible to fix all the detected problems, it is important for design projects to use a thorough kind of problem prioritisation. Such an approach helps to decide which problems are the most important and have to be solved first (Hassenzahl, 2000). An often used method to prioritize usability problems is to determine a severity rating for problems. The severity of a usability problem can be seen as an assessment of the amount of trouble users will experience, as well as a recommendation about allocating the aforementioned resources based on the urgency of fixing problems (Hertzum, 2006). According to Nielsen (1995), severity consists of three factors:

- **Frequency:** How many times does the problem occur? Is it common or rare?
- **Impact:** How hard is it to overcome this problem?
- **Persistence:** Does the problem fade after a number of tries or do users keep on being bothered by it? In other words, do users learn from the previously encountered problem?

From these factors, the severity rate can be determined by using equation 1. As is reflected by the equation, the three factors can be seen as equally important.

$$Severity = frequency * impact * persistence \qquad (1)$$

Even though it is very useful to prioritize usability problems based on severity, it does not get very much attention in the scientific field. Severity is often used in studies as a measure, but the concept of severity itself and how to assess it correctly is almost never studied (Hassenzahl, 2000). The prioritizing of problems in design projects is mostly done by directly assessing severity based on expertise, instead of rating the separate factors and computing these into a severity rating using a clear method or theoretical framework (Hassenzahl, 2000; Hertzum, 2006). As a consequence, one of the factors that define severity, persistence, is more than often neglected in the rating process. For example, Hassenzahl (2000) uses the concept of severity in his study, but does not take persistence into account. Barendregt, Bekker, Bouwhuis and Baauw (2006) did use a longitudinal design in their study, but did not regard persistency as a basic component of severity. This is very concerning as Kjeldskov, Skov and Stage (2010) showed that there is a lot of diversity in the persistence of problems; some problems fade very quickly while others may even remain after a year of extensive use. Most usability studies currently incorporate a cross-sectional study design, which gathers data by measuring at only one point in time (Gerken, Back & Reiterer, 2007). Naturally, by measuring at one point in time it is impossible to accurately look at the persistence of problems. Cross-sectional studies therefore tend to find more usability problems that arise due to a user's first time experience (Gerken et al., 2007). This can lead to results that consist for a big part of 'discoverability' or 'learnability problems' (Gerken et al., 2007; Vaughan et al, 2008). These types of problem may not be as severe as it seems in the long run and provide distorted results (Kjeldskov et al., 2010).

These effects during the first time a product is used seem to be especially true for elderly users. Elderly have typically been found ‘to perform more slowly, to request more assistance during training, and to take longer to acquire computer-based skills’ (Westerman, Davies, Glendon, Stammers & Matthews, 1995, p. 313). However, many of these results on the capabilities of elderly users may show a distorted view as they are based on testing first time experience. Westerman et al. (1995) showed in their study that even though elderly had slower response times at first in an information retrieval task, they approached the faster response times of their younger counterparts after some time. Interestingly, the study showed a massive improvement for the elderly between the first and second trial compared to the younger users, indicating that elderly users were mostly at a disadvantage during early stages of learning. These results show that it is very important look at persistency of problems for elderly users when you want to get a clear image of how well your new product is learned by them, as the elderly need a little bit more time to ‘get started’. This study wants to focus on investigating problem severity and in particular the factor of persistency. Since many studies opt for a use of severity without identifying persistence, it seems important to understand what the consequences are on the results you acquire.

Longitudinality

The best way to study the persistency of problems is to incorporate a longitudinal design. A longitudinal study design differs from the cross-sectional design as it measures on multiple points over time, showing user performance over time (Taris, 2000; Gerken et al., 2007). Changes in the performance with new devices and applications are highly likely to occur and can have a big impact on users (Gerken et al., 2007). The problems that fade quickly may not be worth spending a lot of money on to solve, while the persistent problems are very important to solve as soon as possible. Even though longitudinal designs give you a lot more information about a product, companies are often reluctant to incorporate such a design. Longitudinal studies take a lot more time and as Benjamin Franklin once famously wrote in 1748, ‘Remember, that time is money’ (Franklin, 2004, p. 200). Companies often cannot establish the budget and project schedule capable of measuring the changes over time for problems, so they choose the cheaper solution in the cross-sectional design. Studies have suggested using methods of inspection, such as heuristic evaluation (Nielsen, 1994; Molich & Dumas, 2008) and cognitive walkthrough (Wharton, Rieman, Lewis & Polson, 1994; Molich & Dumas, 2008) , as an alternative to usability testing during certain phases of testing to gather more information for a lower price (Fu, Salvendy & Turley,2002; Hasan, Morris & Probets, 2012). Inspection methods differs from usability testing mostly because it generally does not require any users (Molich & Dumas, 2008). Rather, it asks a small group of usability specialists and domain experts to review the product or service and predict what problems a user will possibly encounter (Fu et al., 2002; Molich & Dumas, 2008; Agarwal & Venkatesh, 2002). Due to the testing without users, this method can be performed in less time and without the effort of finding a lot of participants, helping reduce the costs of the evaluation. We want to build on the use of inspection methods to predict persistency as a cheaper alternative to the longitudinal design. This way companies are able to assess persistency and use this to calculate the severity rating for a problem, while still using the affordable cross-sectional study design.

Contribution of persistency

Beside the main focus of this study to investigate the possibility of predicting problem persistency, this study will also look into the contribution of persistency to severity and usability research. As stated earlier, most studies in the field of usability evaluation opt for a cross-sectional study design and do not gather data about persistency. The consequences of these choices were investigated earlier by Kjeldskov et al. (2010), by looking at the proportion of problems that persisted over a large period of time. They compared the usability problems found by nurses interacting with an ERP system during first time use and after a year of extensive use. The nurses were regarded as novices at the first trial and experts in the last trial.

The level of severity of a problem was assessed by the expert opinion of an evaluator using a three-point scale, as is typical practice (Kjeldskov et al., 2010). The results showed that 40 out of 61 (66%) of all usability problems were still persistent after a year, even though some of them had not been regarded as severe problems during the severity evaluation. All of the three different levels of severity showed problems that were still persistent after a year, which tells us that persistency had a significant impact on severity which was not yet evaluated by the experts. This shows us that persistency is very relevant for determining severity of a problem and therefore in prioritizing problems. An important aspect of these results is that not all problems were found to be persistent. The number of problems that are persistent should be high enough for it to be worth the effort of testing multiple times, but also not too high. If almost every problem had been found to be persistent, it would be impossible to use persistency for prioritisation. If no problem had been found to be persistent, persistency would be irrelevant all together. Reproducing similar results to the study by Kjeldskov et al. (2010) would support the belief that the concept of persistency is important to the field of usability evaluation. As this study wants to make a case for incorporating persistency more often in usability evaluation, we will use the data of this study to replicate the study of Kjeldskov et al. (2010) and will compare the results of both studies. A favourable proportion of persistent problems, comparable to the 66% found by Kjeldskov et al. (2010), will indicate that persistency could contribute to a better form of problem prioritisation.

Learning and persistency

Problem persistency can also be seen as a consequence of the learning that users do or do not show. The learning progress a user makes can be shown with a learning curve. A learning curve depicts how user performance of a task improves with practice, describing the relation between performance of a task and the number of repetitions (Speelman & Kirsner, 2006; Aynalem, 2007). In general a learning curve will start very steep, with dramatic improvements at first, but these improvements gradually taper off with continued practice. However, not all learning curve are the same and different kinds of curves can be expected based on different kinds of learning. Learning a sequential task will often improve gradually, such as for example practicing a piece of music on a piano or a guitar. If you practice to play an instrument you will gradually become faster and make fewer errors during a certain piece. If some does not understand how to perform a task and then discovers the right way all of a sudden, improvement will be very different. The learning curve will in this case be stable at a low level of performance at first and will then increase instantly to a high level. An example of such a case of learning is when someone would be asked to travel to certain location with the train. If he figures out the right trajectory or someone tells him this, that person will not become gradually better in the task of travelling. He will have acquired the necessary knowledge and will be able to perform the task instantly. In some cases, a user will not have a moment of insight or will not find the required information and will not become much better at all. Figure 1 shows illustrations of these different described learning curves. Line a represents a gradual improvement, while line b shows an instant improvement. Line c shows a lack of improvement, demonstrating that a user did not acquire the necessary information or ability.

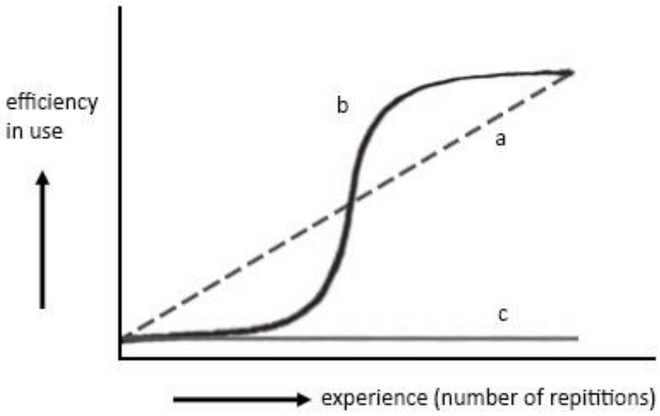

Figure 1. Examples of different learning curves. Line a represents a gradual improvement, line b an instant improvement and line c represents a lack of improvement.

This study will use a discrete approximation of the learning curve to determine persistency of a user for a certain problem. This approximation, consisting of a binary pattern that shows the presence of a problem over a number of trials, will be called a *persistency pattern.* An example of a persistency pattern can be seen in figure 2. As is visible the persistency pattern consists of a value of 'one' or 'zero' which reflects if the problem was detected or not during a certain trial. Note that this means that an increase in ability for a user would eventually lead to a decrease in the persistency pattern. The persistency patterns also reflects the focus in usability research on the problems in a system, rather than the performance of a user.

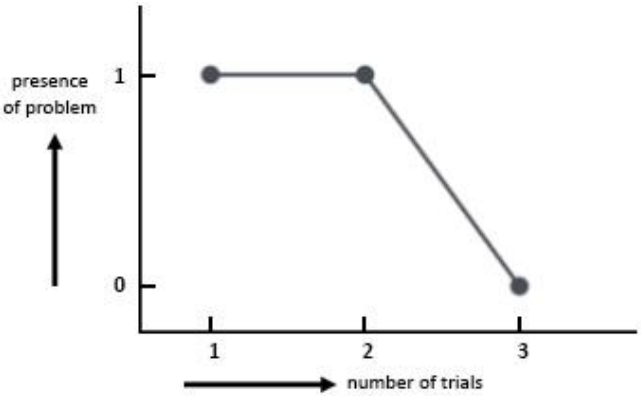

Figure 2. Example of a persistency pattern.

Towards an error classification

The main purpose of this study is to investigate the possible prediction of persistency for different kinds of usability problems. In order to predict problem persistency, we need to make distinctions between usability problems based on certain properties that can be determined in a cross-sectional study. Usability studies have made distinctions between usability problems before based on error classifications. In a study that compared inspection and user testing methods, Fu et al. (2002) proposed that different usability problems occur at different levels of human performance. These levels of human performance were classified using an error classification method by Rasmussen (1983). Barendregt et al. (2006) used a similar error classification method by Zapf, Brodbeck and Prümper (1989) to classify usability problems in a study on identifying usability and fun problems in a computer game. Even though these studies are able to classify usability problems, they did not take persistency of problems into account. This study wants to extend the use of the classification methods and use the error classifications to distinguish between usability problems to predict the persistency of different problems over time. In this study it is hypothesized that different levels of human performance and different classifications for errors could potentially lead to different persistency patterns. To investigate this, the error classifications will be used to classify incidents. Incidents can be seen as separate occasions of errors that the users encounter during interaction with a system. These incidents will then be matched to form usability problems. Matching is the process of grouping problem descriptions together that are similar, to reduce the number of problems to fix. The classifications in this study are different from those that were performed in the studies by Fu et al. (2002) and Barendregt et al. (2006), as these studies classified the usability problems, instead of the incidents. Since the classifications are going to be performed in a different way, an extended matching protocol is necessary for this study.

In a traditional matching protocol, incidents are directly matched to form usability problems. There are a lot of different ways to do this, such as by expert reviews or by using analytical methods (Hornbæk & Frøkjær, 2008). In order to minimize the disagreements between evaluators, which often occur, this study favours the use of a clear and analytical method. By using methods that may help evaluators to anticipate causes and consequences due to their stepwise approach, evaluators will be better in predicting problem handling (Hassenzahl, 2000; Hertzum, 2006; Hertzum, Molich & Jacobsen, 2013). Lavery, Cockton and Atkinson (1997) created such a method that decomposes an incident into four different components: cause, breakdown, outcome, and design change. The decomposition enables evaluators to compare the overlap between multiple properties of an incident. Incidents that have similar characteristics are summarised as a usability problem description.

As an expansion on the existing method of Lavery et al. (1997), this study proposes to use an extended matching scheme that includes a step between the matching of incidents to usability problems, where incidents are classified in different error categories. Figure 3 illustrates all steps that are taken in the extended matching scheme. As can be seen in this figure, the incidents are first classified into error categories, before any matching. After the error classification, incidents are matched using the method by Lavery et al. (1997) and the acquired error categories to create what will be called ‘user errors’. These user errors can be seen as more generally described error description consisting of a group of incidents with the same error classification that also show similarities in their descriptions. Incidents that are the same based on the method by Lavery et al. (1997) that do not have the same error classification can be checked to see if they lead to different user errors, or that an error classification was wrong. Incidents that were classified as unknown at first can also be matched to user errors based on the method by Lavery et al. (1997) and will take on the error classification of the user error. The user errors that are retrieved from the added matching step can then be matched a second time to form usability problems. These usability problems can potentially consist of multiple error classifications, as a usability problem can be encountered as a consequence of various different types of behaviours.

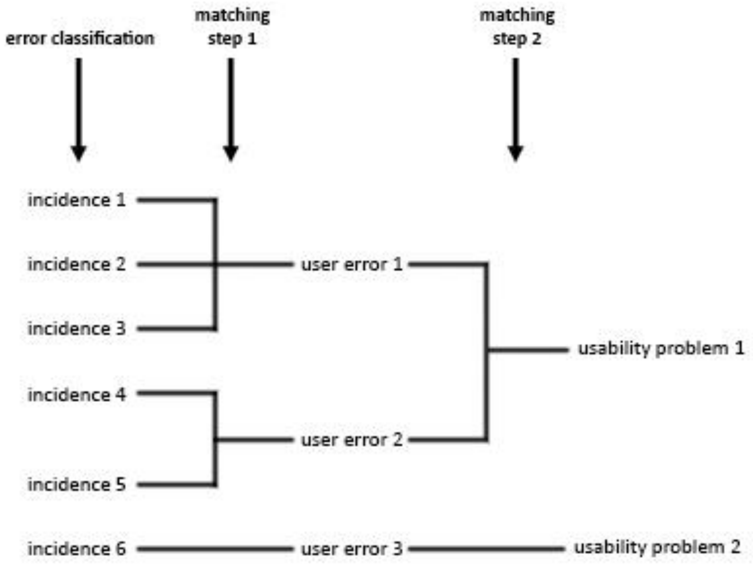

Figure 3. Extended matching protocol.

Error classification methods

To help evaluators classify the incidents to the error categories as described above, we constructed a step-by-step classification guidelines. The classification guideline was based on two existing error classification methods that were used in earlier studies to classify usability problems and show some similarities: the skills, rules and knowledge framework by Rasmussen (1983) and an error classification taxonomy based on action theory by Zapf et al. (1989). Both of these classification methods will be discussed briefly.

Skills, rules and knowledge

Rasmussen (1983) proposed that human performance can either be at a skill-based, rule-based or knowledge-based level. Based on the situation that a user is experiencing, a behaviour can be triggered at different levels of consciousness. Sometimes a stimulus triggers an automatic reaction that a user does not even need to think about, or a user recognises a situation and therefore immediately knows what he has to do. At other times a user will need time to consciously evaluate a complex situation before actually performing an action. The three levels of Rasmussen (1983) reflect these degrees of conscious control exercised by a user (Rasmussen, 1983; Barendregt et al., 2006).

At the lowest level of conscious control, the behaviour of users is skill-based. This level represents specific sensory-motor actions that take place without conscious control and are executed smoothly and automatic. The actions are often based on recognition, meaning that someone recognises a certain type of feedback from the environment and automatically executes a very specific action pattern, which is only suited for a specific purpose (Rasmussen, 1983; Barendregt et al., 2006). At the next level of conscious control, users execute stored sequence of subroutines in familiar situations, started off by a stored rule or procedure. This is called rule-based behaviour. The selection of which rule to apply is often based on previous successful experiences in similar situations. Rule-based behaviour has found to be similar to the use of schemas, which are high-level knowledge structures that support fast processing of routine situations (Besnard & Cacitti, 2005; Reason, 1990). Rule-based performance differs from skill-based performance in the level of conscious attention a user needs to perform an action, but this distinction is sometimes difficult to make. In general, users cannot explain afterwards how they performed skill-based behaviour, while in rule-based performance they can report the rules they applied as an indicative conditional, with an ‘if A then B-like’ structure (Rasmussen, 1983; Barendregt et al., 2006). At the highest level of conscious control, called the knowledge-based level, users perform complex and conscious analyses of situations that are new to them. Feedback from the environment is collected to create an explicit goal. A plan is devised based on this goal and the effect of the plan is tested based on trial and error. At this level, users often need a couple of different tries before the suited action is found. Since users are creating new and complex plans for certain situations with this type of performance, it can be applied at any time and regardless of the prior experience of a user (Rasmussen, 1983; Barendregt et al., 2006). The different levels of performance can also be related to the different kinds of learning curves (review figure 1). Skill-based and rule-based learning often show a gradual improvement due to the needed repetitions of action patterns to perform the task well. This type of learning is reflected by line a in figure 1. Knowledge-based learning complies with the (lack of) acquisition of knowledge necessary to perform a task instantly, as shown in line b or c in figure 1.

Figure 4 shows a simplified visual representation of the behavioural model by Rasmussen (1983) that incorporates the three levels of conscious control. Due to the simplification, the specific processing steps on a level and interactions between the levels were left out. The model by Rasmussen (1983) reflects that when an action is practiced more often and more information is available about the action, shortcuts can be taken to a lower level, enabling a user to process an action faster with less conscious control. The interpretation of stimuli triggers a behaviour at a certain level and help with this process. For example, when someone starts with driving lessons, every action he or she does needs conscious planning and are processed on the knowledge-based level. However, after some time and practice, drivers can control their car almost automatically. They can change gears based on the sound of the engine and adjust their speed immediately if the speedometer shows the car is exceeding the maximum speed, without thinking about how to perform these actions. The stimuli are now processed by taking a shortcut via the skill-based level and the processing itself has become faster. These shortcuts do not necessarily mean that users will be flawless. By getting more experienced and using shortcuts to process information on lower levels of conscious control, you are also increasing the chances of committing errors at a lower level (Frese & Zapf, 1994).

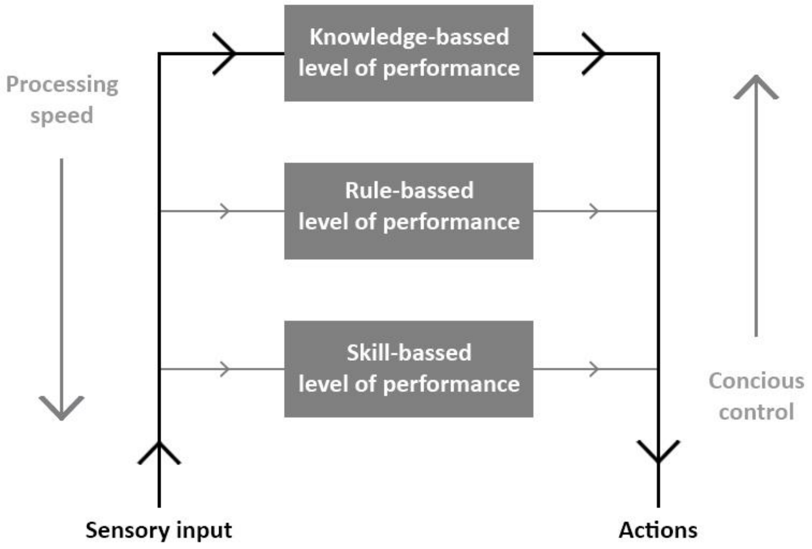
Processing speed
Knowledge-bassed level of performance
Rule-bassed level of performance
Skill-bassed level of performance
Concious control
Sensory input
Actions

Figure 4. Simplified version of the behavioural model by Rasmussen (1983) showing the processing of actions on different levels of behaviour.

Reason (1990) used the framework by Rasmussen (1983) to classify errors in the human performance based on the three levels of conscious control and linked different error types to these levels. Reason (1990) made a distinction between errors that were made intentionally and unintentionally. He stated that on the skill-based level, when an action does not fit the intention of the individual, users can encounter either slips or lapses. Slips are errors that are made when the execution is wrong, while lapses are made when the retrieval of a plan goes wrong, most often due to a failure of memory. Even though slips and lapses are both made unintentional and not affected by the correctness of a plan, they are still very different in detectability. A slip is an easily detectable action that goes wrong, while a lapse is a more covert failure in memory that often does not become an action at all. Users will often be the only one aware of a lapse, while the rest of the environment can become aware of a slip when it occurs.

On the rule-based and knowledge-based level, users can encounter mistakes. Mistakes are inherently different than slips and lapses, because they do happen intentionally. Mistakes are made when a user thinks they are doing the right thing when they actually are not, making mistakes a lot more subtle and complex than slips and lapses. (Reason, 1990; Haar et al., 2013). A mistake is rule-based when it is caused by an intentional but wrong application of a certain rule or assumption. Knowledge-based mistakes occur when a user is in an unfamiliar situation and does not know which rules or actions are fitting to this situation. The user still has an intention, but has no plan or idea on how to accomplish this when starting. Using trial and error the user will try to complete his intention. The difference between rule-based and knowledge-based mistakes can be difficult, but can be made by the presence of the earlier mentioned indicative conditional reasoning statement that is associated with rule-based actions (Rasmussen, 1983; Haar et al., 2013).

Action theory

Another method which can be used to classify behaviour is action theory. This is a behavioural-oriented theory for information processing that tries to analyse actions by looking at regulation and cognitions. Even though Rasmussen did not mention action theory in their work, the similarities are striking and multiple researchers have linked the two methods (Zapf, Brodbeck, Frese, Peters & Prümper, 1992; Frese & Zapf, 1994; Barendregt et al., 2006).

According to action theory, an action can be described from two points of view, namely the action process and the hierarchical structure of the action (Frese & Zapf, 1994). The action process is an iterative process consisting of five steps, from forming a goal and creating a plan, to executing the plan and receiving feedback (Frese & Stewart, 1984). The feedback that is gathered by performing an action can be used to create a new goal, potentially starting the process over again, reflecting the aforementioned iterative nature of the model (Frese & Zapf, 1994). The hierarchical structure reflects the level of conscious regulation by actions using cognitions. The higher levels of regulation are associated with conscious problem solving and have a more flexible, heuristic-like nature, while the lower levels of regulation consist of more rigid algorithmic plans that are fast, processed in parallel, situation-specific and often automatized that are highly stable over time.(Frese & Zapf, 1994; Frese & Stewart, 1984). This structure resembles the skills, rules and knowledge framework by Rasmussen (1983) to a point where they can be seen as almost interchangeable (Frese & Zapf, 1994; Barendregt et al., 2006; Zapf et al., 1992). The biggest difference from the skills, rules and knowledge framework is the addition of the knowledge base for regulation, which does not have an equivalent in the framework by Rasmussen (1983) (Zapf et al., 1992). The knowledge base for regulation can be seen as prerequisites for the regulation processes and as guiding functions in preparing for an action. It consists of at least three aspects that are necessary to regulate actions: knowledge of facts, knowledge of procedures, and understanding in the sense of mental models (Barendregt et al., 2006; Frese & Zapf, 1994).

Zapf et al. (1989) used action theory to create an error classification taxonomy that can be seen in (Zapf et al. 1992). This taxonomy incorporates eight different types of problems, which are classified using the steps of the action process and the hierarchical structure of actions as levels of classification. The five steps from the action process were brought down to three in the taxonomy to make it easier to discriminate between the steps. The new three steps were goals/planning, monitoring and feedback, reflecting the stages before the action, during the action and after the action, respectively. For the skill-based level, the three steps were even combined into one, as it is empirically very difficult to differentiate between steps at this level, since actions are performed without much conscious control (Zapf et al., 1992). The knowledge base for regulation is placed separately in the taxonomy. The equivalents of the skills, rules and knowledge framework (Barendregt et al., 2006) were also added to the corresponding regulation levels in figure 5.

Knowledge base for regulation	Knowledge errors		
Regulation level	Steps in the action process		
	Goals/Planning	Monitoring	Feedback
The intellectual level of action regulation (Knowledge-based)	Thought errors	Memory errors	Judgment errors
The level of flexible action patterns (Rules-based)	Habit errors	Omission errors	Recognition errors
The sensorimotor level of regulation (Skills-based)	Sensorimotor errors		

Figure 5. A taxonomy of errors by Zapf et al. (1989), which classifies errors using action regulation and steps in the action process (Zapf et al., 1992).

The eight different error types that are incorporated in the taxonomy are explained briefly below.

Error types

Knowledge errors: The knowledge base for regulation can be seen as a prerequisite for actions and the error type at this level is related to this concept. Knowledge errors appear when a user does not know the right commands, functions keys or rules in a program. These errors can for example be caused by inadequate instruction about the program or task (Zapf et al., 1992; Barendregt et al., 2006).

Thought errors: Thought errors occur at on the intellectual level during the goals/planning step in the action process. Thought errors occur when the application or service is the reason that users develop inadequate goals and plans, or if the users make wrong plans and sub plans, even though they do have enough knowledge about the system’s functions and features (Zapf et al., 1992; Barendregt et al., 2006). Zapf et al. (1992) give the example for this error of a user who wants to place a 12-columnwide table on a single page, but then finds out the column width has been chosen too wide for it to fit.

Memory errors: Memory errors are found in the taxonomy at the intellectual level during the monitoring phase of the action process. These kind of errors occur when a user forgets to perform a certain part of the plan, even though the plan itself was adequate (Zapf et al., 1992; Barendregt et al., 2006). Zapf et al. (1992) used the creation of table again as an example for this error. They described how a user plans to print a table with a number of labelled columns. After printing, the user sees that one of the planned columns has been forgotten in the table.

Judgment errors: The last error at the intellectual regulation level is the judgment error, occurring during the feedback phase of the action process. When a user does not understand feedback after an action or is unable to interpret it, this is a judgment error (Zapf et al., 1992; Barendregt et al., 2006). Barendregt et al. (2006) give the example of a user that receives feedback on an action during computer game, but is not able to understand from the feedback if the action was right or not.

Habit errors: Habit errors are placed on the level of flexible action patterns in the taxonomy, during the goals/planning step. Habit errors occur when a user executes an action correctly, but in the wrong situation. In other words, an action program was executed that worked in another known situation, but was wrong in this particular situation (Zapf et al., 1992; Barendregt et al., 2006). This type of error can for example occur when users switch to a new program for an old task, or after the redesign of the interface of a known program. The users will try to use the same function keys that they knew from the old situation, but these do not work in the new situation (Barendregt et al., 2006).

Omission errors: Omission errors occur at the level of flexible action patterns during the monitoring step of the action process. This type of error appears when a user does not complete a well-known sub-plan, one they have completed very often. This can be due to the fact that the user is distracted or focussed more on a next step that has to be taken and simply forgets to perform this action (Zapf et al., 1992; Barendregt et al., 2006). Zapf et al. (1992) give the example of a user that forgets to save a file before closing it, even though this is done on a regular basis.

Recognition errors: The third type of error from the level of flexible action patterns, found during the feedback step, is the recognition error. This error occurs when a user does not notice a well-known feedback message or is confused by it (Zapf et al., 1992; Barendregt et al., 2006). It might be important to note that the difference between the recognition error and the judgment error from the intellectual level is that the judgment error has to do with newly received feedback, while recognition errors have to do with interpreting feedback that has been received (and understood) before.

Sensorimotor errors: The last error type is the only one at the sensorimotor level of regulation. The sensorimotor errors are related to the motor-skill that is required to execute an action. Examples of sensorimotor errors are accidentally clicking the wrong mouse button or pressing a button next to the one you planned on pressing.

Error classification guidelines

As stated earlier, this study wanted to create step-by-step error classification guidelines based on a clear and analytical framework to help evaluators classify incidents to an error category and use this classifications for the extended matching scheme. Since the taxonomy by Zapf et al. (1989) can be seen as an expansion on the model by Rasmussen (1983) and has more extensive descriptions of the error categories, the error taxonomy was chosen as the main inspiration for the classification guidelines. However, the classification guidelines did have an extensive introduction that presented the theoretical background of the taxonomy, including the concepts of Rasmussen (1983) and Reason (1990). The classified incidents will be matched in two steps to form usability problems that have a single or mixed error classification assigned to them. These different types of classifications will probably need different types of learning and support for a user to overcome them (Frese & Stewart, 1984; Zapf et al., 1992). As not all problems seem to require the same type of learning, different problems will also show a different persistency pattern. As the persistency patterns are dependent on the learning curves of users, it is also very important to take a look into individual differences.

Individual differences

The persistency patterns consist of the measurements of multiple users, who all show an individual learning curve. As Egan (1988) claimed, ‘differences among people usually account for much more variability in performance than differences in system designs or differences in training procedures’ (Egan, 1988; Freudenthal, 2001), emphasizing the importance to take individual diversity into account. When designing for elderly users, it is might be even more impossible to ‘average’ them as a whole group, since you are designing for a very diverse group with a lot of different backgrounds and possible disabilities that come naturally with old age (Hawthorn, 2003; Shneiderman, 2000). To account for this individual differences, previous experience and technology enthusiasm or geekism will also be used as predictors for the persistency patterns.

Previous experience

According to Hurtienne, Horn, Langdon and Clarkson (2013), previous experience is one of the main factors influencing the performance of older adults with technology, positively influencing the speed and effectiveness of interaction (Czaja & Sharit, 1993; Fisk, Rogers, Charness, Czaja & Sharit, 2009; Langdon, Lewis & Clarkson, 2007; Lewis, Langdon & Clarkson, 2008) and should therefore be taken into account. If users practice more and gain experience, their performance will become better. Most elderly users have very little experience with technical devices and can be seen as novice users. (Hawthorn, 2003). Novice users are known to encounter much more critical and serious problems than experts (Kjeldskov et al., 2010) This means that elderly users with a lot of previous experience are expected to encounter fewer incidents in total than users with less experience.

Geekism

In their effort to investigate differences in users, Schmettow, Noordzij and Mundt (2013) tried to capture a trait of users that is associated with exploring and tinkering with technological devices. They proposed this was caused by a motivational predisposition they called technological enthusiasm, or geekism. They defined geekism as ‘an individual’s strong urge and endurance to understand the inner workings of a computer system’ (Schmettow et al. (2013), p. 2042). They tested this by a variation on the Stroop priming task, showing that subjects with a geek predisposition showed stronger association with geekism words (Schmettow, 2013). It is thought in this study that users with a geek predisposition are going to be motivated to get better at the tasks and are going to try and tinker with all options and try to look for functions beyond the task. Therefore, it is expected that ‘geeks’ will encounter more incidents on the knowledge-based level than the users which do not have this trait.

Study goal and hypotheses

In summary, this study wants to know if the earlier described error classifications can be used to predict the persistency patterns of usability problems. Being able to predict the persistency of usability problems could be a cheaper and easier solution to analytically calculate complete severity ratings for design projects that are not able to implement a longitudinal study design. Of course, users are very different from each other and they do not always show the same learning curve. Therefore, individual differences will also be used as predictors for determining the persistency patterns of the problems. These predictors will be previous experience with the used devices and geekism. The data of the study will also be used to try and prove that measuring persistency is worth the extra effort to obtain useful and relevant information. This will be done by replicating the study by Kjeldskov et al. (2010) and comparing results. We will test the following hypotheses to investigate the mentioned objectives:

H1. Comparable to the study by Kjeldskov et al. (2010), a little over half of the usability problems that are found will be persistent. Such a proportion supports the relevancy of investigating persistency as a part of severity and problem prioritisation.

H2. A higher score on previous experience will lead to fewer encountered incidents in total.

H3. A high score on geekism will lead to a higher number of encountered knowledge-based incidents.

H4. The usability problems encountered on the knowledge-based level and level of knowledge base for regulation will disappears suddenly or not at all. The problem will be persistent, until the knowledge is acquired. Then the persistency pattern will decrease very strongly in a short time. When the knowledge is not acquired the pattern will stay at the same level. Knowledge-based problems and level of knowledge base for regulation will therefore either be persistent or disappear early.

H5. Usability problems at the rule-based level will show persistency pattern that gradually decreases with more experience with the devices. Due to formed habits from earlier trials, it is also possible for problems to appear in later trials. Rule-based problems will therefore either be persistent or appear late.

H6. Usability problems at the skill-based level will show a persistency pattern that gradually decreases with more experience with the devices. Skill-based problems will be mostly persistent.

Methods

Sample

For this test phase of the MOBILE.OLD project a sample was needed with elderly users over 60 years old who were interested in helping the development process of services for elderly. The NFE provided a sample as large as possible, consisting of twenty elderly users from Bunnik, where the main office of the NFE is located, and a care centre in Amersfoort. The NFE used its own network of known elderly and elderly activity groups in the Bunnik area to get participants. In Amersfoort, a supervisor of the care centre helped to find participants by asking the elderly in the care centre if they would be willing to participate. Every participant signed an informed consent form that was provided by the NFE. The participants received a small token of gratitude for their help after the first session was finished.

Unfortunately, the video files of one user (User 7) were corrupted and it was not possible to analyse his video clips. User 7 was therefore removed from the dataset and the data analysis was performed with 19 users between the age of 61 and 82 (*M*=70.68, *SD*=5.38). Six of these users were male, thirteen were female and all users had a Dutch nationality.

Material

Services

MOBILE.OLD prototype applications were developed by a number of technical partners from Greece, Romania and Austria that were all involved in the consortium of the MOBILE.OLD project. Ten types of services were developed in the MOBILE.OLD project in total, which focussed on providing information, physical training support, orientation services and fun. The applications run on Android and were specifically developed for High Definition televisions, smartphones and tablets. All the applications were translated to Dutch before using them in the tests. Every application was only tested on one of the three devices, the one that seemed most suited for the type of service. This division of services over the devices was an agreement between multiple project partners of the consortium, to make sure every testing site would use the same division. The individual services will be described further below:

My Activity (Smartphone): This application uses the GPS to track users when they are hiking, running, cycling or skiing and save these tracks. A track can then be shared with others on the internet.

My Checklist (Tablet): This application can be used to create and edit lists. The items on a list can be checked and unchecked. The items can be added to the list by inserting text, recording an audio message or by making a photo of something (for example of an old packing of something you need from the grocery store). It is also possible to add a (recurring) alarm for a list.

My First Aid (Tablet): This application shares information about how to perform basic first aid. The instructions are shown in text and images and can also be read out loud by the application. The application has a phonebook with all the phone numbers of emergency services in a certain country as well.

My News (Television): This application shares live traffic news and weather reports. Users can find possible delays for travel by car, public transport, airplane and information about the current weather on their location when they use this application.

My Orientation (Smartphone): This application can be used to find the location of points of interest in your direct area. These locations can be saved to the device and a route can be calculated to a location.

My Quiz (Smartphone): This application can be used to create somewhat of a scavenger hunt. The device can be used to (manually or automatically) select locations, which a user needs to find and travel to with the help of a compass that points in the direction of the next location. When users arrive at the correct location, they are asked two questions about the location or the area.

My Safety (Smartphone): This application was made for a more specific group of elderly, who have psychological disorders. A caretaker can use a program to draw safe zones and hazard zones on a map for someone, which is also called ‘geo-fencing’. The application uses the GPS signal to detect if a user is in a safe zone or not. If the user leaves this a safe zone, he/she gets a message that asks the user to return to the safe zone or if they need help. If they do not respond before a certain time, the caretaker gets a message with the last known location of the user.

Due to the different focus of this application and the different user group it targeted, it was later removed from the analyses in this study.

My Training (Television): This application shares exercise videos for physical training and therapy that can be performed by users at home. The exercises can be added to a schedule that can be accessed in the application. When you are scheduled to do an exercise, the system gives you a reminder.

My Trip planner (Television): This application uses Google maps to calculate and/or save a route. After inserting your destination of choice in this application or selecting a saved route, Google maps is started with your route. The navigation of Google maps can then be used to navigate to your destination.

My View (Tablet): This application uses Google Street view to give users a virtual tour of chosen locations. After you choose the location you want to view from a library in the application, Google Street view is launched at the chosen location. This application can be used to view locations someone wants to visit, or to show where you have already travelled.

Questionnaires

A number of questionnaires were used to assess usability as well as the predictors for the classification. All of these questionnaires can be found in appendix A. For assessing user satisfaction, the after-scenario questionnaire (ASQ) (Lewis, 1995; Lewis, 2002) was used. The ASQ has been found to be a good way of assessing subjective usability and was administered after each task during all three trials. It consists of a three scales between one and seven, asking about how satisfied a user is about a certain aspect of a product. Experience was assessed by a questionnaire developed for this study. This questionnaire, consisting of 13 items, focussed on experience with mobile phones and television rather than smartphones and smart TV, as it was expected that almost no elderly user would have had much experience with the newer devices. The score on the experience questionnaire could vary between zero and 36. To assess geekism, a questionnaire by Schmettow et al. (2013) was used. This questionnaire consisted of 19 items that a user had to answer using a six-point Likert scale. The questionnaire also featured a separate category for if the users did not know which answer to choose, which was given the scales mean value of 3.5. The scores of the 19 items were averaged to get one geekism score for each user.

Think aloud procedures

As the screen could not be optimally recorded, it was very important to hear what the participant was thinking and doing. To encourage this behaviour, procedures for thinking aloud were used. These procedures are used to let users verbalise their thoughts and actions to analyse working with the software. The data obtained reflects the actual use of the device and not just the users’ judgment of usability (Ericsson and Simon, 1993). There are two forms of think aloud procedures: concurrent think aloud (CTA) and retrospective think aloud (RTA). While CTA asks users to think aloud while working through the tasks, RTA records all actions by the user and then plays the footage back to the user self. The user can then think-aloud and verbalise his thoughts and actions (Van den Haak, de Jong & Schellens, 2003). For this study, we chose a hybrid form where CTA was used as a default technique, but users could also be asked about their thoughts and actions after the task was done (RTA). The users were not shown the footage of the actions, but the researcher asked them about a certain part of the task directly after the task was done. As RTA takes longer and could bias the users for their next trials, these questions were only asked if something was unclear to the researcher during the testing.

Apparatus

The users performed tests with applications on a Samsung Galaxy Note Smartphone, a Samsung Galaxy Tab 3 and a High Definition television. An Android Set-Top box was connected via HDMI to the High Definition televisions and used to present the applications on the televisions. The Set Top Box was controlled by a wireless Android remote that controlled the mouse pointer on the screen and featured a keyboard for typing in the applications. All testing locations had a Wi-Fi network, which was necessary for the applications.

A user was positioned right in front of the high definition television in a chair with a desk in front of him. The users used the tablet and smart phone at the table and did not have to move from his/her seat to use the Smart TV, making it more comfortable for the users and easier to record the users. One camera was positioned to film the face of the user, while another camera was positioned to film as much as possible of the TV screen, to see what the user was doing. The tablet and smart phone were connected to a laptop, which captured the screen using the program ‘BBQ Screen’. The screen of the laptop was then recorded, saving the screen of the Tablet and Smartphone in a recording. Both camera’s also captured audio, to make sure that all comments and events were registered. All video-capturing devices were connected to Morae on the laptop, a software tool for capturing and editing usability testing material. A layout of the experimental setup can be found in figure 6.

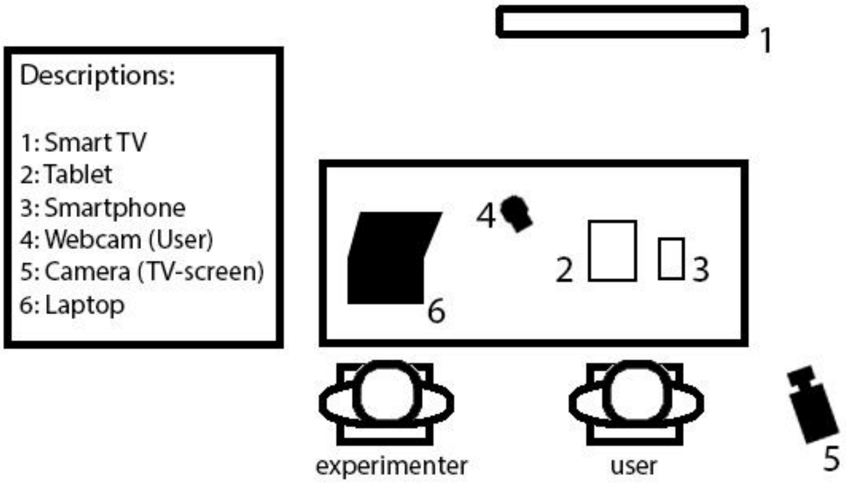

Figure 6. Setup of the experiment.

Procedure

The elderly users were invited to come by the head office of the NFE in Bunnik, a care centre in Amersfoort, or were visited at home. The complete study consisted of two sessions, one week apart. The first session featured the first and second trial and the second session featured only the third trial. When users started the first session, they were first explained about MOBILE.OLD and its goals. They were then presented with an informed consent, a task list and a workbook featuring all the questionnaires for this study. The users were also explained and encouraged to think aloud during tasks, according to the described think aloud procedures. Lastly, the users were asked for permission to be recorded on video for analysis. The workbook was divided in four chapters. The first chapter featured the questionnaires which were used to collect information demographic and individual information of the users. The second, third and fourth chapter featured the ASQ-scales for the first, second and third trials, respectively.

The users started with filling in some personal and demographic questions and the two earlier discussed questionnaires about previous experience and geekism. They could then continue in the workbook when they were finished. The next chapter of the workbook displayed a task number and the corresponding ASQ scale, which had to be filled in after a task was performed. In total, the users had to complete 13 tasks, divided over the nine different applications. When a user thought that the task was finished, or if he had the feeling that he would not get any further with the task, he was asked to fill in the questionnaire and continue to the next page, featuring a new task. When the second chapter featuring the tasks was started, the recording devices were turned on. Two cameras were used, one filming the face and hands (thus interaction with the touchscreen) and the other filming the activity on the TV-screen. The program BBQ screen recorded the touchscreen, so no camera was needed to record this. After all tasks were finished, the PSSUQ and a market survey were also filled in by the user. The results of these questionnaires were not used in this study, but were provided to the MOBILE.OLD project, as they requested.

Once the users completed the first trial, they got a 15-minute break to drink some coffee or tea. It was the aim in this study to keep the time limited to a maximum of 90 minutes for each trial, to take guidelines for running experiments with elderly users into account (Barrett & Kirk, 2000; Lines & Hone, 2004). After the break, the users were asked to start on the next chapter of the workbook, which featured the second trial. They were asked to repeat the 13 tasks and fill in the related ASQ scales. After completion of the second trial, the respondents had finished the first session.

After approximately a week, the users were again invited or visited for the second session and the third trial was performed. The users repeated the tasks for the third and last time and filled in the ASQ scales, completing the entire workbook. The users were thanked for their cooperation and were given a voucher as a token of gratitude when they had finished the second session.

Data Gathering & Analysis

The video recordings were analysed using Morae, looking for incidents. When something happened in a recording that needed to be reviewed, a marker was placed in Morae. Almost 3000 markers were placed and investigated, leading to a total of 1424 incidents. These incidents were then classified using newly created guidelines based on the work of Zapf et al. (1989) & Zapf et al. (1992). After obtaining the classifications, the incidents were matched to form user errors and usability problems, respectively.

Problem classification guidelines

The problem classification guidelines were created to help with classifying user errors and categorising them in the categories from the model by Zapf et al. (1992). This made it possible to further inspect the qualities of errors assigned to categories and to make predictions about new errors and usability problems. The guidelines were created to be a step-by-step guide on classifying the errors in the categories. To support the use of tablets and pc’s, Hyperlinks were put in the text to help with the navigation of the guidelines. An overview of the steps in the classification guidelines can be seen in figure 7.

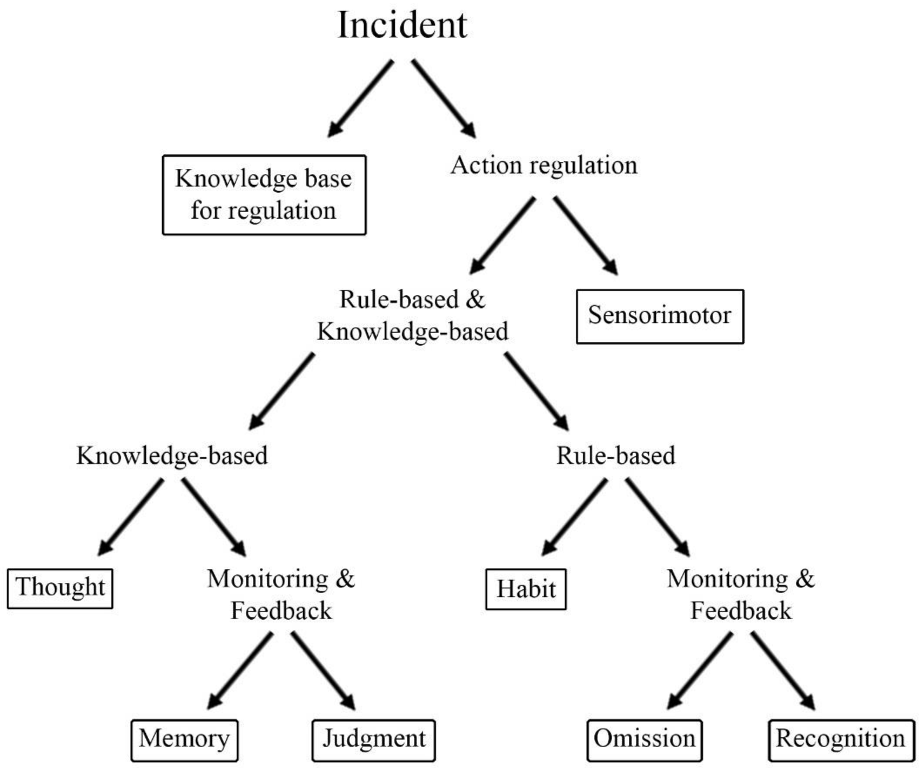

Figure 7. Overview of the steps that are taken in the classification guidelines. Every step consists of a choice between two alternatives until an error category is reached. These error categories were marked by a rectangular box.

The guideline first distinguishes the knowledge errors from the seven other action-based errors. The distinction between these user errors was made using the definition of these categories given by Zapf et al. (1992), Frese & Zapf (1994) and Barendregt et al. (2006). For the action-based errors, the regulation level is determined next. This distinction was made by using the work of Haar et al. (2013), who made a model on how to classify problems in the skills, rules and knowledge framework by Rasmussen (1983) which as mentioned before is very similar to the concept of regulation level in the model by Zapf et al. (1989). After the regulation level is clear, the progression in the action process is determined. This was again done by using the definitions on the categories given by Zapf et al. (1992), Frese & Zapf (1994) and Barendregt et al (2006). The guidelines consisted of seven choices that eventually led to all the eight categories from the taxonomy by Zapf et al. (1989). Each step consisted of a number of 'relevant questions' for the choice that had to be made between two options. The relevant questions could be answered by 'yes' or 'no' and the answers were connected to one of the two options to go forward in the steps. When a choice led to a final category, control statements and examples were given to do a final check on the category. When it was unclear to what category an error needed to be assigned, errors were put in the unknown category. Approximately 10% of the incidents were assigned to the unknown category. The error classification guidelines that was used can also be found in appendix B.

Matching process

The incidents were first matched to create user errors using the method by Lavery et al. (1997) (Hornbæk & Frøkjær, 2008; Haar et al., 2013), and the error classifications. The method by Lavery et al. (1997) consists of dividing and describing the incident in a number of components such as context, cause, breakdown and outcomes (Lavery et al., 1997). By extensively describing the incident, it is possible to compare multiple aspects of the incident among one another. For the matching in this study, incidents were combined when they resembled each other very much in multiple categories and combined each other in the error classification that was determined by the guidelines. Incidents in the unknown error category were also reviewed, to see if incidents that did have a classification resembled incidents in the unknown category on multiple components. If possible, the incidents from the unknown category were added to a user error. Since this study wanted to find as many different errors as possible, user errors that were represented by just one user were still included, other than some studies choose to do (Fu et al., 2002; Følstad, Law & Hornbæk, 2012). As advised by Følstad, Law and Hornbæk (2012), these errors were checked to see if they were artefacts from the test situation. If this was true, the errors were disregarded and removed from the dataset.

The user errors found by matching the incidents, were matched a second time to obtain usability problems during the second matching step. The user errors which were thought to be caused by the same design issue, were grouped together. This second matching phase was based on the similar changes method (Hornbæk & Frøkjær, 2008). This method consisted of grouping the user errors that are thought to be caused by the same design issue in a usability problem. By fixing a design issue, all the user errors that were grouped in the corresponding problem would be resolved as a result. During the matching using the similar change method, the cause, breakdown and design solution categories from the previous matching phase were still used as heuristics for finding common design issues. Some of the user errors could not be matched, because they directly led to a separate new usability problem. Others user errors were found to be caused solely by a behavioural component and could not be matched to a usability problem.

These user errors, which accounted for 4% of the total number of user errors, were therefore disregarded and were not represented in the obtained usability problems Figure 3 can be reviewed for a visual representation of the described steps in the extended matching scheme. As is visible in this figure, the situation where one user error leads to multiple usability problems is disregarded in this study.

The error classifications were used as input to form usability problems, but it often happened that a usability problem consisted of user errors from different error categories. This can happen when user errors are caused by the same design issue, but this design issue leads to different kinds of behaviour. To be able to still see the error classification for the usability problems, a different approach was chosen. The number of incidents in a certain error classification that were matched to a usability problem, were divided by the total of incidents matched to that usability problem. This was done to acquire the percentages of how many incidents that were added to a usability problem via user errors were classified as a skill-based, rule-based, knowledge-based, unknown or knowledge base for regulation error. The regulation levels were not further specified in the steps of the action process, as some error categories were almost unused. Each percentage was used as a predictor in the statistical model during analysis.

Binary coding

We made a detection matrix for usability problems for each separate trial. By combining the three matrices from each trial, it was possible to create a binary code reflecting the presence of a problem over the three trials for a certain user. These binary codes all reflected a specific persistency pattern, but some of these patterns reflected the same trend. The codes were combined to create three groups, each consisting of three binary codes reflecting the same trend for a problem:

- **Group 1:** Problem appears early, then disappears (100, 010, 110)
- **Group 2** Problem appears in later trial (001, 011, 010)
- **Group 3** Problem is persistent over time (011, 111, 101)

Five algorithmic-like rules were created to help to assign the binary codes to the right groups. These rules can be found in appendix C. The code ‘000’ was not assigned to any of the groups, as learning is not present for users in these cases. Users with this code were already able to use the service without encountering the problem.

The three groups reflected trends of problems that are interesting to designers. The persistent problems are the most interesting to designers and should always get the highest priority in the next redesign phase. Problems that appear in later trials could be caused by flaws in the advanced options that users encounter when they are more acquainted with the services, or due to wrongly formed habits. Not all users will encounter these problems, as not all users will be interested in advanced functions. However, these problems are still interesting to the designers, as the advanced functions will let you use the service to the full extent. These problems should therefore be a significant priority during re-design, but problems to the basic functions of the application should always get a higher priority. Problems appearing early and then disappearing can be often disregarded by designers if resources are not available.

Contribution of persistency

This study also used its data to investigate a sub goal about the relevance of testing persistency. The testing of persistence can only be useful if the proportion of persistent problems is not too large and not too small, as this would both make persistency irrelevant to problem prioritization. The relevancy of persistency was tested by replicating the study by Kjeldskov et al. (2010). As discussed earlier, Kjeldskov et al. (2010) tested a user group twice, with the sessions a year apart. They looked at how many usability problems were gone, how many had persisted and which problems were newly encountered. These three groups were compared to the disappear early, persistent and appear late trends from this study, respectively. To compare the results from this study to the study by Kjeldskov et al. (2010), the data was structured in the same manner. Kjeldskov et al. (2010) disregarded usability problems that were only found by one user. To account for this, only the usability problems found by more than one user in a certain trial were used for this comparison. As a consequence, seven usability problems were not taken into account, meaning that the total number of usability problems that were used was 42. Kjeldskov et al. (2010) divided their usability problems into three different categories: critical problems, serious problems and cosmetic problems. As this study did not use such a division, all problems from the study by Kjeldskov et al. (2010) were computed into a single score.

Results

Various sorts of data were collected for the MOBILE.OLD project, but these will not all be discussed in this paper. In accordance with the study goal to investigate problem persistency, this results section will focus on the comparison of the current study with the study by

Kjeldskov et al. (2010), the error classifications and the persistency patterns. The analyses for the discovery rates for the user errors and problems, the ASQ scores for satisfaction and time on task scores can be found in appendix D. A list of all the found usability problems can be found in appendix E.

Contribution of persistency

The results of the study by Kjeldskov et al. (2010) were compared to the scores of users in the current study after the third trial. The presence of a problem at first use were compared to the presence of a problem after a year of extensive or in the third trial of the study by Kjeldskov et al. (2010) or the current study, respectively. Figure 8 shows the number of usability problems for the three different levels of severity by Kjeldskov et al. (2010) as seen in their article. These three levels of severity were 'critical problems', 'serious problems' and 'cosmetic problems'.

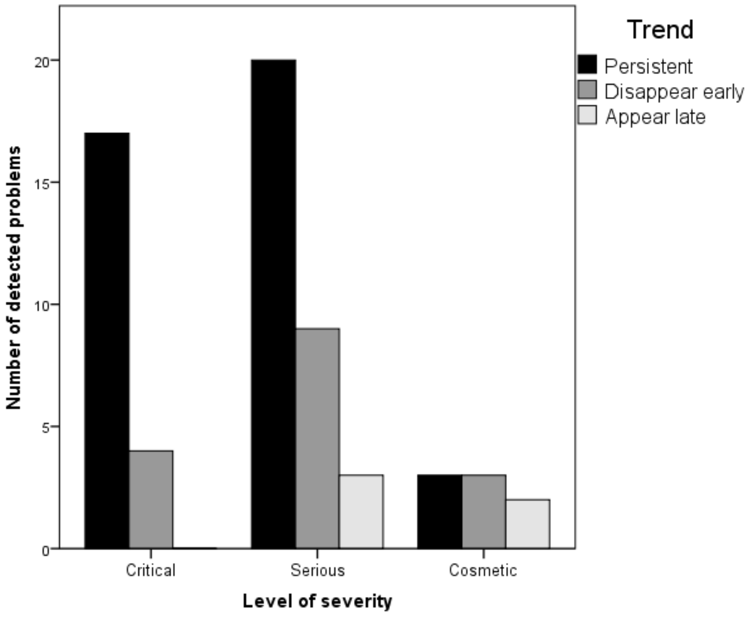

Figure 8. Representation of persistency in the study by Kjeldskov et al. (2010).

These three severity categories were summed up to gain one score to be able to compare with the current study, which meant that 40 of the problems were found to be persistent, 16 disappeared after a year of use and 5 appeared late. In this current study 32 problems persisted over three trials, 6 problems disappeared before the third trial and 4 appeared later than the first trial. A visual representation comparing the totals of both studies can be found in figure 9. The studies did not have the same number of usability problems to compare, therefore percentages were given instead of values. As is visible from this figure, the studies show comparable results for all three trends. The biggest difference is that the current study shows more persistent problems than the study by Kjeldskov et al. (2010) with 67% to 76%, respectively. The percentages still seem to be relatively alike, or at least of the same order of magnitude.

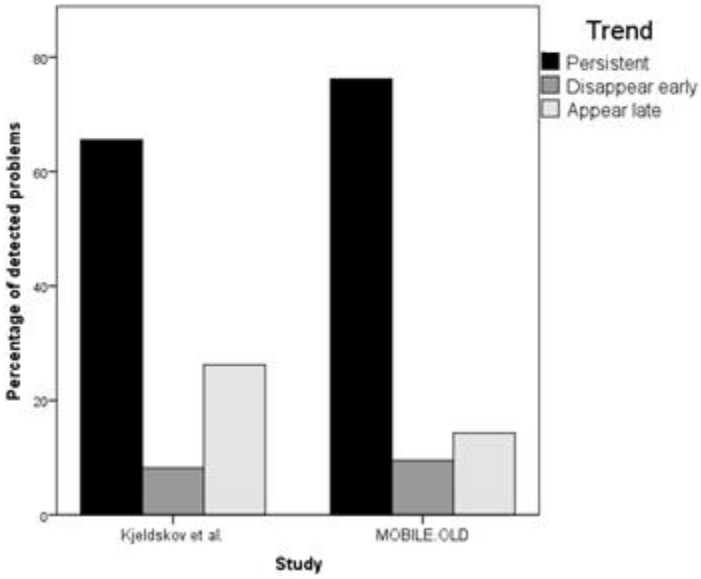

Figure 9. Comparison of problem persistency in the study by Kjeldskov et al. (2010) and the current study.

Error classifications

During the use of the classification guidelines, a number of cases were encountered that were difficult to classify. These cases were written down to be able to learn from them and improve the guidelines. The incidents in these cases were either classified as unknown or a choice was made for all these kind of cases and a reasoning was written down. The distribution of the different error types during each of the steps in the matching scheme can be seen in figure 10. This figure shows the percentages of the total number of classifications for incidents, user errors and usability problems. The differences between the incidents and user errors came from the unknown incidents that could be matched to a user error with incidents that did have a known error classification, leading to a greater number of specified classifications. The usability problems sometimes consisted of mixed classifications, so the percentages of error classifications also differed somewhat from the incidents and user errors. The percentages for the usability problems were still based on the number of incidents in a certain user error, to account for the frequency of incidents in a user error. The percentage of memory problems was set to 0%, because the user errors with a memory error classification could not be matched to a usability problem.

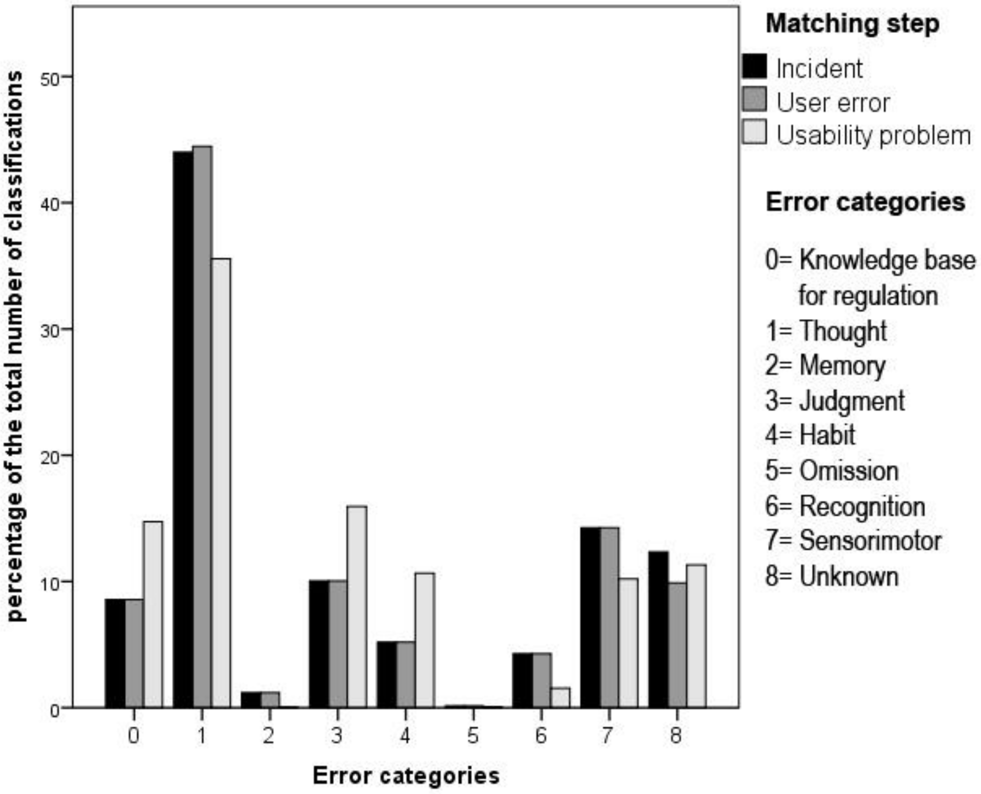

Figure 10. Distribution of the eight different classifications and the unknown category for incidents, user errors and problems.

Figure 10 shows that we found almost no memory problems, omission problems and recognition problems. Due to the uneven distribution of error types, we decided to form a new division of classifications for the analyses that would possibly lead to clearer results. Five error types were used in further analyses: knowledge base for regulation (henceforth abbreviated to KBFR), knowledge-based, rule-based, skill-based and unknown. This division resembles the original framework by Rasmussen (1983), but still uses the KBFR separately. This was done because a significant percentage of the problems received a knowledge error classification. This new division of classifications can be seen in figure 11.

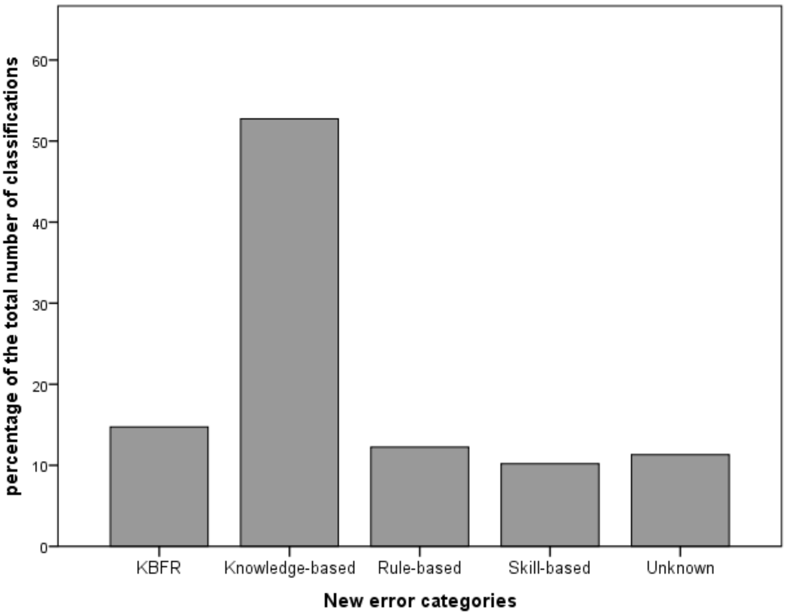

Figure 11. Distribution of new classifications for usability problems.

During the testing, one of the five covariates that explained the error classifications in a problem became redundant in the statistic model due to an unknown reason. This was always the last covariate in the model in the SPSS editor window. To account for this, it was decided to remove one of the covariates from the model, which neutralised this effect. As the unknown classification was the only classification that was not necessary to accept or reject the hypotheses, this classification was removed from the model.

Individual differences

Descriptive statistics were calculated for previous experience (*M*=22. 18; *SD*=5.07) and geekism (*M*=3.78; *SD*=0.62) to look at the distribution of scores. The scores on previous experience showed a rather large standard deviation, showing that there was a lot of variation in experience level in the sample. The scores on geekism showed a mean score more than one standard deviation above the average score of the scale, showing that the elderly that participated scored pretty high on the geekism trait on average. To further investigate the influence of these individual differences on usability problems, the number of incidents a user encountered for a problem, which was called incident frequency, was used as a dependent variable for a generalized estimated equations analysis. The age of the user and the score on the previous experience questionnaire were both considered as predictive measures for previous experience. Older users were thought to have had a lesser exposure to technical devices in their jobs and pastimes, making them less experienced with technical devices. The data was checked for violations of the assumptions that are necessary for the chosen statistical using the protocol by Zuur, Leno and Elphick (2010), which can be found in appendix F. These checks showed the possible presence of overdispersion. As the dependent variable consisted of count data that could only take on integer values a negative binomial distribution was used. This was chosen in favour of the often used Poisson model, due to the violations in the assumptions for that statistical model. To account for possible effects of learning and fatigue, autoregressive generalized estimated equations were performed. The results of the analysis can be found in table 1 and the syntax for the regression in appendix G.

Table 1
Results from generalized estimated equations with Poisson distribution for previous experience

			95% Confidence Interval			
Parameter	Beta	Standard Error	Lower bound	Upper bound	Significance	QICC
Intercept	5.127	1.895	1.414	8.841	0.007	421.509
Previous experience	0.021	0.112	-0.198	0.241	0.850	
Age	0.017	0.027	-0.035	0.069	0.519	
Previous experience * age	-0.001	0.002	-0.004	0.002	0.670	
Geekism	-0.028	0.067	-0.159	0.104	0.679	
Geekism * KBFR	0.343	0.134	0.081	0.605	0.010	
Geekism * knowledge-based	-0.120	0.083	-0.283	0.043	0.150	

Note. Due to an unknown reason, one of the classification covariates always became redundant in the statistic model. Therefore, the unknown classification was removed as a covariate from the model.

The main effect for both previous experience and age and the interaction effect between previous experience and age can be seen at the top of the table. No significant results were found for these predictors. The results showed very low beta values for all predictors and the confidence intervals around zero made it impossible to view the direction of the effects. The results also showed no main effect for geekism or interaction effect for geekism and the percentage of knowledge-based problems, with low beta values and no clear direction of the relation shown by the confidence intervals. However, there was a significant interaction effect found for geekism and the percentage of knowledge base for regulation problems. The positive beta value of .343 indicated that users with a high score on geekism encountered a more incidents that were classified as KBFR than users with a low score on geekism. Exponentiation of the beta value showed that an increase of one unit on geekism led to a raise of 1.4 in the number of encountered incidents.

Persistency patterns

Disappear early

We chose a generalized estimated equations analysis for investigating the persistency patterns as well, with the presence of a disappear early trend as the dependent variable. The data was checked again for outliers and violations of homogeneity and normality, to see if the analysis would be appropriate (Zuur et al., 2010) and showed a violation of the normality assumption. These checks can be found in appendix F. As stated by Zuur et al. (2010), linear regression is relatively robust against such a violation and transforming the data could make differences harder to detect. Therefore it was chosen to not transform the data and use the generalized estimated equations regardless of the violation. Due to the binary dependent variable, a binary logistic distribution was chosen with an autoregressive model to account for possible learning effects and effects of fatigue. The results for the percentages of KBFR and Knowledge-based classifications are shown in table 2. The syntax for the regression can be found in appendix G.

Table 2
Generalized estimated equations for problems that disappear early

			95% Confidence Interval			
Parameter	Beta	Standard Error	Lower bound	Upper bound	Significance	QICC
Intercept	0.725	3.972	-7.059	8.509	0.855	920.886
Knowledge-based	-0.961	0.591	-2.119	-0.197	0.104	
KBFR	-0.040	0.190	-0.412	0.332	0.833	
Rule-based	-1.390	0.350	-2.076	-0.704	0.000	

Note. Due to an unknown reason, one of the classification covariates always became redundant in the statistic model. Therefore, the unknown classification was removed as a covariate from the model.

The table shows that no main effects for were found for knowledge-based problems and KBFR problems on the trend to disappear early. The beta values are large enough to indicate an effect, but the confidence intervals are too wide to show certainty for a direction of the effect or a relation at all. The results did indicate a significant negative effect for rule-based problems, indicated by the beta value of -1.390. Exponentiation of the beta value for a binary logistic value gives us the odds ratio of .87, which means that the odds of disappearing early for a rule-based problem are 1.14 times smaller (1/.87=1.14) than the odds for other problems types. This effect would indicate that a lower number of rule-based problems will disappear early compared to other problems, but this is not a very large effect.

Appear late

With the presence of the appear late trend selected as a dependent variable, the generalized estimated equations were performed once again. As can be seen in appendix F, the assumptions were once again checked. This showed that the data violated the same normality assumption as the disappear early pattern showed. Despite the violation, the generalized estimated equations were again deemed to be robust enough to account for the violation (Zuur et al., 2010) and used. Due to the binary dependent variable, a binary logistic distribution was chosen again with an autoregressive model to account for possible learning effects and effects of fatigue. The results for the percentages of rule-based classifications are shown in table 3. The syntax for the regression can be found in appendix G.

Table 3
Generalized estimated equations for problems that appear late

			95% Confidence Interval			
Parameter	Beta	Standard Error	Lower bound	Upper bound	Significance	QICC
Intercept	-2.051	6.115	-14.036	9.935	0.737	848.018
Rule-based	0.276	0.362	-0.433	0.986	0.445	

Note. Due to an unknown reason, one of the classification covariates always became redundant in the statistic model. Therefore, the unknown classification was removed as a covariate from the model.

The results did not show any significant effects for the trend to appear late. The results showed a confidence interval around zero, making it impossible to indicate if there is a relation between the rule-based classification and the appear late trend.

Persistent

Generalized estimated equations were performed a last time, after the data was checked on statistical assumptions (Zuur et al., 2010). These checks can be found in appendix F. Besides the presence of heteroscedasticity and some outliers, no violations were found. Due to the binary dependent variable, we chose a binary logistic distribution here as well with an autoregressive model to account for possible learning effects and effects of fatigue. The results for percentages of error classifications are shown in table 4. The syntax for the regression can be found in appendix G.

Table 4
Generalized estimated equations for problems that are persistent

			95% Confidence Interval			
Parameter	Beta	Standard Error	Lower bound	Upper bound	Significance	QICC
Intercept	4.619	6.499	-8.119	17.356	0.477	402.409
Skill-based	0.102	0.569	-1.014	1.217	0.858	
Rule-based	-0.172	0.965	-2.062	1.719	0.859	
Knowledge-based	0.492	2.321	-4.058	5.042	0.832	
KBFR	0.951	2.713	-4.367	6.269	0.726	

Note. Due to an unknown reason, one of the classification covariates always became redundant in the statistic model. Therefore, the unknown classification was removed as a covariate from the model.

As visible in table 4, there were no significant effects found for the persistent trend. Even though the beta values often indicated an effect, the confidence intervals were too large to interpret the direction of the relations or if there was a relation at all.

Discussion

Findings

Contribution of persistency

The replication of het study by Kjeldskov et al. (2010) showed a very comparable level of problem persistency in both studies and hypothesis H1 will therefore be accepted. The level of persistency was a little higher in the current study, but this was not surprising. The difference between time that users had to interact with the system in the study by Kjeldskov et al. (2010) and the current study would explain this difference. Users in the study by Kjeldskov et al. (2010) had a whole year to work with the ERP system and still 66% of the problems persisted, compared to 76% of the problems over three trials in the current study. It seems that testing three times will give you an adequate approximation of an expert level performance in these kind of situations.

As the results of both studies are very much alike, the conclusions by Kjeldskov et al. (2010) will translate to these results as well. Both studies support the idea that it is important to investigate the persistency of problems, as some problems will not fade away just by becoming more experienced with a system or service. The percentage of persistent problems in both studies show that it should be worthwhile to investigate the persistency. It will help developers in prioritizing their problems, as about 25% of their total number of problems could receive a lower priority with regard to persistency. Prioritization can of course not be done based only on these kind of numbers of persistency, as the frequency and impact would then be disregarded. However, the numbers from both studies support the idea that persistency is an equally important component of severity as frequency and impact, since persistency was able to discriminate between problems in a way that frequency and impact could not.

Individual differences

No relation was found between the previous experience and the number of encountered incidents, which means that the hypothesis H2 will be rejected. This does mean that no relation between previous experience and the number of encountered incidents can exist. It may be that previous experience with technical devices was not relevant for the users, as the applications differed from other devices so strongly that all elderly users were novices. If the previous experience was irrelevant, it would make sense to find no relation between previous experience and the number of incidents. The number of encountered incidents for each different trial was also not taken into account in the results for previous experience. This may be of influence on the results of this study.

The results showed only a relation for geekism with knowledge base for regulation (KBFR) incidents but no effect for knowledge-based incidents, so hypothesis H3 can be partially accepted. Users with the geek trait were hypothesised to be very curious of new features and would try out functions that they did not yet know. It was also hypothesised that the same curiosity would also lead to more knowledge-based problems, but this was not the case. Every user encountered mostly knowledge-based problems when faced with a new device, not just the geeks.

Persistency patterns

The results of this study showed no relation between KBFR problems and the disappear early trend. Likewise, no relation was found between the knowledge-based problems and the disappear early trend. For this reason hypothesis H4 is rejected. Something that did became apparent is that the knowledge-based problems accounted for a very large part of the total number of problems compared to other classifications. This large proportion, more than half of all classifications, could indicate that knowledge-based problems are the main category that developers and designers should target when trying to support elderly users during their first interactions with technical devices, as this is the most common error classification.

Furthermore, no significant effects were found for the relation between rule-based problems and the appear late and persistent trend as well. The analysis for the disappear early trend did find a significant negative effect for rule-based problems. If rule-based problems are less often found to disappear early, this is in support of the hypothesis H5 that rule-based problems either appear late or are persistent. However, this effect was so small that the hypothesis H5 will still be rejected.

Hypothesis H6 could not be accepted, as there was no significant relation found for skill-based problems and the persistent trend. The hypotheses H4, H5 and H6 also incorporated expectations about the persistent pattern, but no relations were found for any of them. It seems that interacting or mediating variables should be sought after to understand the different circumstances that lead to a certain classified problem to be persistent.

Relevant implications

The results of this study have several relevant implications for usability research. This study set out to provide researchers with an easier and possibly cheaper way to assess persistency for severity ratings, as well as to show that these efforts are worth it by showing the relevancy of persistency to severity. It proved difficult to predict the persistency of the problems that elderly encountered during their learning efforts based on error classification. We found no insightful relations between the persistency patterns and the classifications, so there is no proof at this point that it is possible to predict persistency using error classifications. Despite not finding possibilities to predict persistency, we found persistency itself to be a very useful concept to usability research. The comparison between the current study and the study by Kjeldskov et al. (2010) showed that the results of both studies are in favour of measuring persistency in usability research. Persistency was able to make distinctions in prioritization that a severity rating based on frequency and impact was unable to do. If resources are available, it is valuable to use a longitudinal design early in the project to help with determining the severity of problems. The costs of alterations to your product in a later stage of development are known to be vastly higher (Boehm & Basili, 2005) than early stages, so this can make the extra effort in early stages worthwhile in the long run.

The significance of persistency to usability research also has implications for the much debated subject in usability testing of how many subjects are necessary for a successful usability evaluation. A lot researchers believe that four or five users should be sufficient during testing to find up to 85% of the possible usability problems and that this percentage should be enough. This idea is based on the equation

$$Found(i) = N\left(1 - (1 - \lambda)^{i}\right) \qquad (2)$$

where N is the total number of problems, 'λ' is the average probability of problems found while testing a single user and i is the number of evaluators or users (Nielsen & Landauer, 1993; Nielsen, 2000; Schmettow, 2012). However, a number of others studies have discussed that this view is too simplified and breaks a number of basic assumptions from the used binominal model, such as completeness and homogeneity (Lewis, 2001; Schmettow, 2008; Schmettow, 2012). One of the assumptions that does not seem to comply with the current study is that the problems found by an evaluator or user are independent of whether they were found before and independent of each other. It is difficult to assume that the findings are independent from each other if they were found earlier by the same person in an earlier trial. It may well be that there is different probability for finding a problem in a second or third trial. If the probability for finding a certain problem would remain the same, this would mean that one user tested three times is worth the same as three users. If there is no additional detection for a single user at all during later trials, this mean that multiple trials are irrelevant for the detection rate of usability problems. Of course, the truth will be somewhere in the middle, but it may be interesting to investigate this further and to test the appropriateness of equation 2 (Nielsen & Landauer, 1993) for longitudinal designs.

The use of user errors helped well in classifying the incidents and matching them, so it is recommended to keep on using and improving the extended matching protocol. Beside the improvement of the matching protocol, the guidelines can also be improved by learning from the cases that were difficult to classify with the guidelines and other difficulties that were experienced during use. The distribution of error classifications that were obtained by using the guidelines showed that the elderly users mostly found problems on the knowledge-based level of performance during their interactions with the new devices. This is consistent with other studies, such as the study by Fu et al. (2002) and Kjeldskov et al. (2010), who found that novice users encounter more knowledge-based problems than expert users. Fu et al. (2002) proposed that this difference in encountered problems is caused by an incorrect mental model of the tasks. Barendregt et al. (2006) found somewhat different results when testing children between 5 and 7, as these children found mostly KBFR problems. Both the knowledge-based problems and KBFR problems are associated with the early learning stages, so the results found by Barendregt et al. (2006) are still comparable, but this shows that the young children and elderly users may not be seen as the same, novice user group. This study agrees with Barendregt et al. (2006) that it is possible that further use of the services and experience will lead to a greater proportion of the problems made in lower levels of performance, as is often assumed (Finstad, 2008), However, elderly users clearly need support with knowledge-based behaviour to get started, especially when keeping in mind that elderly are often scared of trying new technological products (Eastin & LaRose, 2006). Designers and developers should learn from this and need to focus the support functions of their devices on knowledge-based problems for the elderly users. These support functions should be easily and explicitly available, as the first encounters with a new product can be crucial for elderly users in determining whether they would want to use it again in the future. It may be that the capabilities and wishes of elderly users will change over the coming years. As the population is growing older, the ‘new’ elderly will be a group with much more experience with computers and smartphones due to their exposure in their working career, so the use of smartphones among the elderly will surely increase in the not too distant future. However, specialized designs will be remain necessary to support this age group and problem severity is very useful in the process of developing these designs.

It is not clear if the found results on persistency and error classifications can be generalized, since elderly users have been marked as a specific user group (Hawthorn, 2003; Shneiderman, 2000). Some studies have argued that adults learn vastly different from children and adopted the concept of androgyny, which differentiates between the learning processes of these two age groups (Purdie & Boulton-Lewis, 2003; Knowles, 1996). This is also supported by the described differences in the number of knowledge-based problems and KBFR problems found by children in the study by Barendregt et al. (2006) and by elderly in the current study. Purdie and Boulton-Lewis (2003) stated that it may be most productive to look at the learning needs of users of different ages, rather than the age itself. This may also be true for the learning of elderly users, as they will probably have other learning needs than their younger counterparts. For example, Purdie and Boulton-Lewis (2003) found in their study that transportation was one of the most important learning needs for the elderly, as they had often lost their driving license. This is a need that would be very different in other age groups. Learning needs may not only be found to be different for age groups, but also for other subgroups in the population. The concept of learning needs could show to be good addition to predictors for the persistency patterns besides the individual traits that users show.

Study limitations

A number of limitations were experienced during this study. First of all, the preparations did not make it able to test the reliability and validity of the previous experience questionnaire, so it was not possible to conclude if the results for previous experience questionnaire were true reflections of the experience level that users had, or measured something else that was similar. Second, various reasons led to missing data. User 7 was removed from the dataset due to corrupted video files, technical difficulties with the applications such as bugs and system errors led to the removal of some tasks for certain users and the time slot for the first session, incorporating trial 1 and trial 2, proved to be too short for some users to perform all tasks. In total, 7% of the tasks were missing from the final dataset. Third, the unknown classification, which accounted for almost 12% of the classifications, had to be removed from the statistic model as an unknown occurrence led to the redundancy of a covariate for the generalized estimated equations. As the elderly users encountered mostly knowledge-based problems, it was difficult to make strong arguments about the error types that were not found often. Fourth, the classification guidelines that were used were not used before. A number of cases were encountered that were difficult to classify. These cases were registered to be able to use them to improve the guidelines. Last, an important limitation for the results section was the overlap in use of binary codes for the trends. For example, the code 010 was used in the appear late trend as well as in the disappear early trend. As some cases were used more than once, it could have potentially lead to overrepresentation of significant effects. However, even with this overlap no clear results were found for predicting the trends, so the limitation did not lead to different conclusions.

Future research

The results of this study indicate that predicting problem persistency based on error classifications is not very promising. Predicting persistency based on the intuition of a usability expert may deliver better results than the use of the classification guidelines, but this would have to be researched before any definitive conclusions about predicting persistency can be made. Even though the prediction of persistency did not show any real results, persistency itself was proven to be very relevant to the field of usability testing. We were able to make distinctions between problems based on persistency that frequency and impact could not. We strongly advise developers to consider using longitudinal designs more often and to use the retrieved results as input for an objectively calculated severity rating. As stated, it seems necessary to investigate how well problem persistency can be predicted by expert intuition before any definitive conclusions about predicting persistency can be made.

This study also found some more general results that can inspire future research in the field of usability evaluation. Future studies on error classification may want to develop the extended matching protocol further, as this did seem to work rather well in classifying all the incidents. The guidelines that were used to classify the incidents as part of the extended matching protocol can be improved by using the experiences and difficult cases from this study to become easier in use and more thorough. It may also be useful for future studies on learning to look at the learning needs that users have based on individual differences, instead of just an individual feature (Purdie & Boulton-Lewis, 2003). Using other individual differences that have been found to influence learning in other studies as predictors may also be investigated. Examples of such predictors are working memory (Wang, Ren, Altmeyer and Schweizer, 2013) and emotions (Mega, Ronconi & De Beni, 2014). It would be possible to classify the tasks instead of the incidents as well to see if task difficulty influences the type of problems that follow based on the required behaviour to complete the task. Last, it may be very interesting to investigate the influences of longitudinal designs on the sample size that is appropriate for usability testing.

Conclusions

The aim of this study was to find a relation between persistency patterns of problems and error classification, to able to use this relation to predict the persistency of problems over time. The possibility to predict the persistency of problems could help other studies and design projects to calculate a more objective severity rating for problems in a less time consuming way. The acquired results did not make it possible to obtain this aim. We found no real evidence suggesting that error classifications could be used to predict the persistency patterns. Despite the efforts of creating the classification guidelines and the extended matching protocol, this still did not lead to any meaningful findings. However, the sub goal of showing the relevance of persistency by replicating the study by Kjeldskov et al. (2010) was successfully achieved, as the results showed that 76% of the problems in this study persisted over three sessions. Such a number is large enough to make an extra measurement worth the extra resources, as well as small enough to make prioritization based on persistency possible.

The elderly users proved to be very able of using the new technological devices and the developed applications. Appendix D show more information about the capabilities of the elderly during testing and how satisfied they were with the applications. In general, the results were in line with the findings by Westerman et al. (1995), as the elderly became much better in performing after the first trial. They were also very satisfied in general about the services and their own capabilities. Furthermore, the distribution of classifications showed that elderly encountered mostly knowledge-based problems. Since the elderly were identified as novice users, this came as no surprise. This distribution urges designers and developers that focus on the elderly as a user group to focus on fixing knowledge-based problems prior to other types of problems and to make sure that their services have a very good support function to help elderly acquire the necessary knowledge for functions. Users that scored high on geekism were found to encounter KBFR problems more often, which supports the aspect of ‘geeks’ being very curious about things they do not yet know (Schmettow et al., 2013). As there were some pretty high scores on geekism despite the age of the users, the personality trait is something that seems to transcend age and could be regarded as a very general aspect of personality.

In conclusion, we did not find predicting persistency based on error classification to be possible, but the concept of problem persistency itself proved to be very useful for determining severity. Measuring severity to its full extend remains important, as human error will always be ‘inevitable, even when straightforward tasks are performed by experienced users’ (Kay, 2007, p. 442; Hollnagel, 1993). If budget is available or can be made available, a longitudinal study design should always be preferred, as there is always the possibility for problems to appear late or disappear early. The most important thing for design companies is that they should be aware of problem persistency and the impact that it has on severity. Even though predictive measures can reduce costs during testing, it will always be scientifically more favourable to be able to see a phenomenon compared to predicting it. In the long run, it may even pay off more to know how problems are going to behave over time, because you can find problems in a system early in design.

References

Agarwal, R., & Venkatesh, V. (2002). Assessing a Firm's Web Presence: A Heuristic Evaluation Procedure for the Measurement of Usability. *Information Systems Research*, *13*(2), 168–186. doi:10.1287/isre.13.2.168.84

Ambient assisted living joint programme (2012). *MOBILE.OLD.* Retrieved 11 February 2015 from www.aal-europe.eu/projects/mobile-old/

Aynalem, S. (2007). Two stage analysis of learning curves on laparoscopic study of surgeons. Retrieved from http://hdl.handle.net/1942/3405

Barendregt, W., Bekker, M. M., Bouwhuis, D. G., & Baauw, E. (2006). Identifying usability and fun problems in a computer game during first use and after some practice. *International Journal of Human-Computer Studies*, *64*(9), 830–846. doi:10.1016/j.ijhcs.2006.03.004

Bargh, J. A., & McKenna, K. Y. A. (2004). The internet and social life. *Annual Review of Psychology*, *55*, 573–90. doi:10.1146/annurev.psych.55.090902.141922

Barrett, J., & Kirk, S. (2000). Running focus groups with elderly and disabled elderly participants. *Applied Ergonomics*, *31*(6), 621–629. doi:10.1016/S0003-6870(00)00031-4

Besnard, D., & Cacitti, L. (2005). Interface changes causing accidents. An empirical study of negative transfer. *International Journal of Human-Computer Studies*, *62*(1), 105–125. doi:10.1016/j.ijhcs.2004.08.002

Boehm, B., & Basili, V. (2001). Software defect reduction top 10 list. *Computer*, *34*(1), 135–137. doi:10.1109/2.962984

Carroll, J. M., & Rosson, M. B. (1987). Paradox of the active user. In J. M. Carroll (Ed.), *Interfacing Thought* (pp. 80–111). Cambridge, MA: MIT Press. Retrieved from http://portal.acm.org/citation.cfm?id=28451

Centraal Bureau voor de Statistiek (2012). *Ouderen beginnen pas op latere leeftijd te vereenzamen.* Retrieved 14 May 2014 from http://www.cbs.nl/nl-NL/menu/themas/dossiers/levensloop/publicaties/artikelen/archief/2012/2012-ouderen-vereenzaming-dns-pub.htm

Czaja, S. J., & Sharit, J. (1993). Age differences in the performance of computer-based work. *Psychology and Aging*, *8*(1), 59–67. doi:10.1037/0882-7974.8.1.59

Eastin, M. S., & LaRose, R. (2006). Internet Self-Efficacy and the Psychology of the Digital Divide. *Journal of Computer-Mediated Communication*, *6*(1). doi:10.1111/j.1083-6101.2000.tb00110.x

Egan, D. E. (1988). Individual differences in human-computer interaction. In M. G. Helander (Ed.), *Handbook of human-computer interaction* (pp. 543–568). Amsterdam, The Netherlands: North Holland publishing company.

Ericsson, K. A., & Simon, H. A. (1998). How to Study Thinking in Everyday Life: Contrasting Think-Aloud Protocols With Descriptions and Explanations of Thinking. *Mind, Culture, and Activity*, *5*(3), 178–186. doi:10.1207/s15327884mca0503_3

Finstad, K. (2008). Analogical Problem Solving in Casual and Experienced Users: When Interface Consistency Leads to Inappropriate Transfer. *Human-Computer Interaction*, *23*(4), 381–405. doi:10.1080/07370020802532734

Fisk, A., Rogers, W., Charness, N., Czaja, S., Sharit, J. (2009). *Designing for older adults: principles and creative human factors approaches*, Boca Raton: CRC Press.

Følstad, A., Law, E. L., & Hornbæk, K. (2012). Outliers in usability testing. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction Making Sense Through Design - NordiCHI '12* (p. 257). New York, New York, USA: ACM Press. doi:10.1145/2399016.2399056

Franklin, B. (2004). *Franklin: The Autobiography and Other Writings on Politics, Economics, and Virtue*. (A. Houston, Ed.). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511806889

Freudenthal, D. (2001). Age differences in the performance of information retrieval tasks. *Behaviour & Information Technology*. doi:10.1080/0144929011004974

Frese, M., & Stewart, J. (1984). Skill Learning as a Concept in Life-Span Developmental Psychology: An Action Theoretic Analysis. *Human Development*, *27*(3-4), 145–162. doi:10.1159/000272909

Frese, M., & Zapf, D. (1994). Action as the core of work psychology: A German approach. In H. C. Triandis, M. D. Dunnette, & L. M. Hough (Eds.), *Handbook of Industrial and Organisational Psychology, Vol. 4* (pp. 272–340). Palo Alto, CA: Consulting Psychologist press. Retrieved from http://www.researchgate.net/publication/232492102_Action_as_the_core_of_work_psychology_A_German_approach/file/d912f5089061de6386.pdf

Fu, W. T., & Gray, W. D. (2004). Resolving the paradox of the active user: Stable suboptimal performance in interactive tasks. *Cognitive Science, 28*(6), 901–935. doi:10.1016/j.cogsci.2004.03.005

Fu, L., Salvendy, G., & Turley, L. (2002). Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology*, *21*(2), 137–143. doi:10.1080/02699050110113688

Gerken, J., Bak, P., & Reiterer, H. (2007). Longitudinal evaluation methods in human-computer studies and visual analytics. In *Visualization 2007: IEEE Workshop on Metrics for the Evaluation of Visual Analytics*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.1024&amp;rep=rep1&amp;type=pdf

Haar, A., Schmettow, M., & Schraagen, C. (2013). *Developing of a Qualitative Classification Method for Usability Errors after Rasmussen*. University of Twente.

Hasan, L., Morris, A., & Probets, S. (2012). A comparison of usability evaluation methods for evaluating e-commerce websites. *Behaviour & Information Technology*, *31*(7), 707–737. doi:10.1080/0144929X.2011.596996

Hassenzahl, M. (2000). Prioritizing usability problems: data-driven and judgement-driven severity estimates. *Behaviour & Information Technology*, *19*(1), 29. doi:10.1080/014492900118777

Hawthorn, D. (2003). How universal is good design for older users? In *Proceedings of the 2003 conference on Universal usability - CUU ’03* (pp. 38-45). New York, New York, USA: ACM Press. doi:10.1145/957205.957213

Hertzum, M. (2006). Problem Prioritization in Usability Evaluation: From Severity Assessments Toward Impact on Design. *International Journal of Human-Computer Interaction*, *21*(2), 125–146. doi:10.1207/s15327590ijhc2102_2

Hertzum, M., Molich, R., & Jacobsen, N. E. (2013). What you get is what you see: revisiting the evaluator effect in usability tests. *Behaviour & Information Technology*, (April 2013), 1–19. doi:10.1080/0144929X.2013.783114

Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, *64*(2), 79–102. doi:10.1016/j.ijhcs.2005.06.002

Hornbæk, K., & Frøkjær, E. (2008). Comparison of techniques for matching of usability problem descriptions. Interacting with Computers, 20(6), 505–514. doi:10.1016/j.intcom.2008.08.005

Hurtienne, J., Horn, A.-M., Langdon, P. M., & Clarkson, P. J. (2013). Facets of prior experience and the effectiveness of inclusive design. *Universal Access in the Information Society*, *12*(3), 297–308. doi:10.1007/s10209-013-0296-1

ISO (2008). Ergonomics of human-system interaction — Part 171: Guidance on software accessibility (ISO 9241-171). London: International Standards Organization.

Kay, R. H. (2007). The role of errors in learning computer software. *Computers & Education*, *49*(2), 441–459. doi:10.1016/j.compedu.2005.09.006

Kjeldskov, J., Skov, M. B., & Stage, J. (2010). A longitudinal study of usability in health care: does time heal? *International Journal of Medical Informatics*, *79*(6), 135–43. doi:10.1016/j.ijmedinf.2008.07.008

Knowles, M. (1996). Andragogy: An emerging technology for adult learning. In R. Edwards, A. Hanson, & P. Raggatt (Eds.), *Boundaries of adult learning* (pp. 82–98). London: Routledge.

Langdon, P., Lewis, T., & Clarkson, J. (2007). The effects of prior experience on the use of consumer products. *Universal Access in the Information Society*, *6*(2), 179–191. doi:10.1007/s10209-007-0082-z

Lavery, D., Cockton, G., & Atkinson, M. P. (1997). Comparison of evaluation methods using structured usability problem reports. *Behaviour & Information Technology*, *16*(4-5), 246–266. doi:10.1080/014492997119824

Lewis, J. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. International Journal of Human-Computer Interaction, 7(1), 57-78. doi:10.1080/10447319509526110

Lewis, J. R. (2001). Evaluation of Procedures for Adjusting Problem-Discovery Rates Estimated From Small Samples. *International Journal of Human-Computer Interaction*, *13*(4), 445–479. doi:10.1207/S15327590IJHC1304_06

Lewis, J. R. (2002). Psychometric Evaluation of the PSSUQ Using Data from Five Years of Usability Studies. *International Journal of Human-Computer Interaction, 14*(3-4), 463–488. doi:10.1080/10447318.2002.9669130

Lewis, T., Langdon, P. M., & Clarkson, P. J. (2008). Prior Experience of Domestic Microwave Cooker Interfaces: A User Study. In P. Langdon, J. Clarkson, & P. Robinson (Eds.), *Designing Inclusive Futures*. London: Springer London. doi:10.1007/978-1-84800-211-1

Lines, L., & Hone, K. S. (2004). Eliciting user requirements with older adults: lessons from the design of an interactive domestic alarm system. *Universal Access in the Information Society*, *3*(2), 141–148. doi:10.1007/s10209-004-0094-x

Mega, C., Ronconi, L., & De Beni, R. (2014). What makes a good student? How emotions, self-regulated learning, and motivation contribute to academic achievement. *Journal of Educational Psychology*. doi:10.1037/a0033546

Molich, R., & Dumas, J. (2008). Comparative usability evaluation (CUE-4). *Behaviour & Information Technology*, *27*(3), 263–281. doi:10.1080/01449290600959062

Nielsen, J., 1994, Heuristic evaluation. In J. Nielsen & R. Mack (Ed.), *Usability Inspection Methods* (pp. 25-62). New York: John Wiley and Sons, Inc.

Nielsen, J. (1995, January). Severity Ratings for Usability Problems. Retrieved January 12, 2015, from http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

Nielsen, J. (2000, March). Why you only need to test with 5 users. Retrieved November 10, 2014, from http://www.useit.com/alertbox/20000319.html

Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In ACM Press (Ed.), *Proceedings of INTERCHI 1993* (pp. 206–213). New York, NY, USA: ACM Press. doi:10.1145/169059.169166

Portio Research Limited (2013). *Mobile Application Futures 2013-2017*. Retrieved 16 January 2015 from http://www.portioresearch.com/en/mobile-industry-reports/mobile-industry-research-reports/mobile-applications-futures-2013-2017.aspx

Purdie, N., & Boulton-Lewis, G. (2003). The learning needs of older adults. *Educational Gerontology*. doi:10.1080/713844281

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-13*(3), 257–266. doi:10.1109/TSMC.1983.6313160

Reason, J. (1990). *Human Error*. New York, NY: Cambridge University Press.

Rice, M., & Alm, N. (2008). Designing new interfaces for digital interactive television usable by older adults. *Computers in Entertainment*, *6*(1), 1. doi:10.1145/1350843.1350849

Schmettow, M. (2008). Heterogeneity in the usability evaluation process. In *BCS-HCI '08: Proceedings of the 22nd British HCI Group Annual Conference on HCI 2008* (pp. 89–98). Swinton, UK: British Computer Society.

Schmettow, M. (2012). Sample size in usability studies. *Communications of the ACM*, *55*(4), 64. doi:10.1145/2133806.2133824

Schmettow, M., Noordzij, M. L., & Mundt, M. (2013). An implicit test of UX. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13* (pp. 2039–2048). New York, New York, USA: ACM Press. doi:10.1145/2468356.2468722

Shneiderman, B. (2000). Universal usability. *Communications of the ACM*, *43*(5), 84–91. doi:10.1145/332833.332843

Speelman, C. P., & Kirsner, K. (2006). Transfer of training and its effect on learning curves. *Tutorials in Quantitative Methods for Psychology*, *2*(2), 52–65. Retrieved from http://www.doaj.org/doaj?func=abstract&id=693682

Taris, T. (2000), *Longitudinal data analysis*. London: Sage publications.

Tomaka, J., Thompson, S., & Palacios, R. (2006). The relation of social isolation, loneliness, and social support to disease outcomes among the elderly. *Journal of Aging and Health, 18*(3), 359–84. doi:10.1177/0898264305280993

Van den Haak, M., De Jong, M., & Schellens, P. (2003). Retrospective vs. concurrent think-aloud protocols: Testing the usability of an online library catalogue. *Behaviour & Information Technology*, *22*(5), 339–351. doi:10.1080/0044929031000

van't Veer-Tazelaar, P. J., van Marwijk, H. W. J., Jansen, A. P. D., Rijmen, F., Kostense, P. J., van Oppen, P., … Beekman, A. T. F. (2008). Depression in old age (75+), the PIKO study. *Journal of Affective Disorders*, *106*(3), 295–9. doi:10.1016/j.jad.2007.07.004

Wang, T., Ren, X., Altmeyer, M., & Schweizer, K. (2013). An account of the relationship between fluid intelligence and complex learning in considering storage capacity and executive attention. *Intelligence, 41*(5), 537–545. doi:10.1016/j.intell.2013.07.008

Westerman, S. J., Davies, D. R., Glendon, A. I., Stammers, R. B., & Matthews, G. (1995). Age and cognitive ability as predictors of computerized information retrieval. *Behaviour & Information Technology*, *14*(5), 313–326. doi:10.1080/01449299508914650

Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The Cognitive Walkthrough Method: A Practitioner's Guide. In J. Nielsen & R. Mack (Ed.), *Usability Inspection Methods* (pp. 105-140). New York: John Wiley and Sons, Inc.

Zapf, D., Brodbeck, F., & Prümper, J. (1989). Handlungsorientierte Fehlertaxonomie in der Mensch-Computer Interaktion. *Zeitschrift Für Arbeits-Und Organisatiepsychologie, 33* (4), 178-187.

Zapf, D., Brodbeck, F. C., Frese, M., Peters, H., & Prümper, J. (1992). Errors in working with office computers: A first validation of a taxonomy for observed errors in a field setting. *International Journal of Human-Computer Interaction*, *4*(4), 311–339. doi:10.1080/10447319209526046

Zuur, A. F., Ieno, E. N., & Elphick, C. S. (2010). A protocol for data exploration to avoid common statistical problems. *Methods in Ecology and Evolution*, *1*, 3–14. doi:10.1111/j.2041-210X.2009.00001.x

Appendix A: Questionnaires

General and demographic Questions

Participanten nr. MOBILE.OLD:	*Dit wordt ingevuld door de onderzoeker*	
Voornaam, Achternaam:		
Datum van Sessie:		
Geslacht:		
Leeftijd:		
Voormalig beroep:		
Huishoudelijk inkomen:	500 – 1000 € 1000 – 2000 € 2000 – 3000 € > 3000 € Geen antwoord	
Hoe groot is uw ervaring met technologie?		
Erg hoog: Ik gebruik regelmatig (minstens twee keer per week) mijn PC en Mobiele telefoon of tablet om op internet te gaan. Hoog: Ik gebruik alleen mijn PC om op het internet te gaan en doe dit één of twee keer per week. Gemiddeld: Ik he been PC en internet, maar ik gebruik dit bijna niet. Ik heb ook niet de neiging om dit meer te gaan gebruiken. Laag: Ik heb geen PC of internet en heb nog nooit of zelden dit soort technische apparaten gebruikt.		
Hoe staat u tegenover technologie?		
Positief: Ik vind het niet erg om nieuwe apparaten uit te proberen als ik ze krijg. Neutraal: Ik weet het niet zo goed of het maakt me niet veel uit. Negatief: Ik vind technologie niets voor mij en blijf er ver van uit de buurt.		

Technical Affinity (Geekism) questionnaire	Ik ben er helemaal mee eens	Ik ben er mee eens	Ik ben er een beetje mee eens	Ik ben er een beetje niet mee eens	Ik ben er niet mee eens	Ik ben er helemaal niet mee eens	Weet ik niet
1 Ik wil begrijpen hoe computer(onderdelen)/Software werken.							
2 Als er iemand hulp nodig heeft met computers, probeer ik zo goed mogelijk te helpen.							
3 Privacy(-instellingen) op de computer of internet zijn er belangrijk voor mij.							
4 Gecompliceerde taken met technische apparaten schrikken mij af.							
5 Ik heb al eens technische apparaten gebruikt voor dingen waarvoor ze niet bedoeld zijn, of ze aangepast.							
6 Objectiviteit is belangrijk voor mij.							
7 Ik heb niet het gevoel veel controle over mijn technische apparaten te hebben.							
8 In mijn vrije tijd breng ik niet meer tijd door met computers/technische apparaten dan met andere mensen.							
9 Wanneer ik nieuwe technische producten koop is het technische vermogen belangrijker voor mij dan het uiterlijk.							

	Ik ben er helemaal mee eens	Ik ben er mee eens	Ik ben er een beetje mee eens	Ik ben er een beetje niet mee eens	Ik ben er niet mee eens	Ik ben er helemaal niet mee eens	Weet ik niet
10 Het motiveert mij technische apparaten te optimaliseren of aan te passen aan mijn wensen.							
11 Ik heb al eens projecten/werkstukken van mij gratis online gezet of zou dit doen.							
12 Ik denk dat er mensen zijn die mij computerfreak zouden noemen.							
13 Het binnenwerk van technische apparaten en/of het programmeren van software interesseert mij niet.							
14 Ik vermijd de geavanceerde opties van mijn technische apparaten.							
15 Ik vind het leuk mijn projecten en ideeën met andere mensen te delen.							
16 Uitdagende opgaven met de computers prikkelen mij.							
17 Ik heb veel kennis over computers (Hardware/Software).							
18 Ik probeer dingen zo wetenschappelijk mogelijk te benaderen.							
19 Ik ben geïnteresseerd in technische producten die meervoudig inzetbaar zijn.							

Technical experience Questionnaire

Technische Ervaring

De volgende vragen gaan over uw ervaring met technische apparaten, zoals Smart TV's, computers, Mobiele telefoons of tablets. Omcirkel of kruis het rondje aan met het antwoord wat voor u het meest van toepassing is.

1) **Ik ben in het bezit van een Smart TV**

Ja / Nee

2) **Ik heb wel eens gewerkt met een Smart TV (dit kan ook bij andere mensen thuis zijn, bijvoorbeeld bij familie of vrienden).**

Ja / Nee

3) **Ik ben in het bezit van een Mobiele telefoon**

Ja / Nee

4) **Ik heb wel eens gewerkt met een Mobiele telefoon.**

Ja / Nee

5) **Ik ben in het bezit van een computer, laptop of tablet.**

Ja / Nee

6) **Ik heb wel eens gewerkt met een computer, laptop of tablet.**

Ja / Nee

Als u hierboven ‘Ja’ heeft ingevuld, gelieve dan de volgende 2 vragen ook in te vullen. Als u ‘Nee’ heeft geantwoord, ga door naar vraag 7.

a) Waar heeft u gewerkt met een computer/laptop, tablet of Mobiele telefoon? (meerdere antwoorden zijn mogelijk)

O - In mijn huis

O - Bij familie thuis

O - Bij vrienden thuis

O - Op het werk

O - Anders, namelijk: ___________________________________

b) Als u op het werk heeft gewerkt met een computer/laptop, Mobiele telefoon of tablet, gebruikte u deze dan voor het grootste deel van uw werkzaamheden?

O - Ik gebruikte geen computer op mijn werk.

O - Ik gebruikte de computer voor een klein deel van al mijn werkzaamheden.

O - Ik gebruikte de computer voor ongeveer de helft van al mijn werkzaamheden

O - Ik gebruikte de computer voor meer dan de helft van al mijn werkzaamheden

7) **Heeft uw wel eens op het Internet gezeten?**

Ja / Nee

8) **Heeft u wel eens een e-mail naar iemand verstuurd?**

Ja / Nee

Hierna volgen een aantal vragen over uw gemiddelde gebruik van technische apparaten. Vul hier een getal in of kruis het antwoord aan wat het beste bij u past.

9) **Hoeveel uren per week werkt u gemiddeld met een Smart TV?**

(denk hierbij aan het aantal uren per week in de afgelopen maand).

____

Als u nog nooit met een smart TV hebt gewerkt, zet dan een 0

10) **Hoeveel ervaring zou u zelf zeggen dat u heeft met een Smart TV?**

O - Ik heb geen ervaring met een Smart TV

O - Ik heb een beetje ervaring met een Smart TV

O - Ik heb een gemiddelde ervaring met een Smart TV

O - Ik heb een bovengemiddelde ervaring met een Smart TV

O - Ik ben zeer ervaren met een Smart TV

11) **Hoeveel uren per week werkt u gemiddeld met een Mobiele telefoon?**

(denk hierbij aan het aantal uren per week in de afgelopen maand).

____

Als u nog nooit met een Mobiele telefoon hebt gewerkt, zet dan een 0

12) **Hoeveel ervaring zou u zelf zeggen dat u heeft met een Mobiele telefoon?**

O - Ik heb geen ervaring met een Mobiele telefoon

O - Ik heb een beetje ervaring met een Mobiele telefoon

O - Ik heb een gemiddelde ervaring met een Mobiele telefoon

O - Ik heb een bovengemiddelde ervaring met een Mobiele telefoon

O - Ik ben zeer ervaren met een Mobiele telefoon

13) **Hoeveel uren per week werkt u gemiddeld met een computer, laptop of tablet?**

(denk hierbij aan het aantal uren per week in de afgelopen maand).

____

Als u nog nooit met een computer, laptop of tablet hebt gewerkt, zet dan een 0.

14) **Hoeveel ervaring zou u zelf zeggen dat u heeft met een computer, laptop of tablet?**

O - Ik heb geen ervaring met een computer, laptop of tablet

O - Ik heb een beetje ervaring met een computer, laptop of tablet

O - Ik heb een gemiddelde ervaring met een computer, laptop of tablet

O - Ik heb een bovengemiddelde ervaring met een computer, laptop of tablet

O - Ik ben zeer ervaren met een computer, laptop of tablet

ASQ scales

ASQ Vragenlijst voor taak

Over het geheel genomen ben ik tevreden met het gemak waarmee ik de taak kan voltooien.

Helemaal mee oneens								*Helemaal mee eens*

Over het geheel genomen ben ik tevreden met de tijd die het me gekost heeft om de taak te voltooien.

Helemaal mee oneens								*Helemaal mee eens*

Over het geheel genomen ben ik tevreden met de ondersteunende informatie (help-functie, berichten op het scherm en andere documentatie) die ik kreeg om de taak te voltooien.

Helemaal mee oneens								*Helemaal mee eens*

Appendix B: Guidelines

Guideline for classifying Usability problems

Introduction

Usability problems can be very useful when done correctly; In most cases when a product is being developed, multiple rounds of usability tests are performed to find and fix as many problems as possible before the product is being launched into the real world market. However, sometimes a user makes a certain mistake when working with a product for the first time, learns, and thus does not make the same mistake when working with the system later on. Knowing which usability problems will solve themselves over time, as explained before, and which ones will stay can save a lot of time and money: Problems can be classified in less rounds of usability testing and there needs to be done less work about fixing found usability problems as just those problems that are classified as “will stay over time” need specific focus to be fixed and those that solve themselves do not.

This document contains a basic guideline for classifying usability problems accordingly. First, the theory behind this usability classification guideline will be explained shortly, consisting of a mix from previous classification theories. Second, there will be some ideas on how to process your data before you can use this classification guideline properly. Next, there is a detailed description of the two dimensions and the eight types of problems of which the guideline consists, along with multiple control questions and examples to compare with your own set of usability problems.

Theory

Theory by Reason

Classifying mistakes has been done previously, as this explains a lot about how the human brain works. To start out with, Reason (1990) stated two basic types of mistakes:

1) **Execution failures – consisting of slips and lapses:** Here, a user knows what to do (intention or plan is correct) but the execution is not. Logically, this only happens

in situations which are known for the user. The difference between a **slip** and a **lapse** is that:

a) A **slip** concerns a situation in which the execution was incorrect.
b) A **lapse** concerns a situation in which there was no execution at all.

2) **Planning failures**: A user does not know what to do (the intention or plan is incorrect) with a rather logical consequence that the execution is incorrect as well (most of the time, sometimes users take a good guess). These mistakes occur at settings that are rather unknown for the user.

Theory by Rasmussen

This basic classification can then be compared to aspects of another classification model that was created by Rasmussen (1983). In his model, Rasmussen defines that there are multiple dimensions of regulation control, or how conscious the action patterns are that the users express. This model contains the following, ranging from highest level of conscience to lowest level of conscience (“automatic” behaviour):

1) Knowledge based behaviour and mistakes: At this level, plans are made and regulation is mostly conscious, so there are no automatic processes but rather a serial step-by-step way of thinking and applying rules (like when following a step-by-step manual for putting together furniture from IKEA). A mistake will mostly belong to this category when a user has no rules known to the situation. Therefore, mistakes in this category are very diverse. A known cause is often an overload of information in a (too) short amount of time.

2) Rule based behaviour and mistakes: At this level, there is a lower level of conscious processing than at the intellectual level. Processing is mostly done in schemata by using ready-made programs which have to be specified by parameters and only work in certain situations (for example when you know how to bake a basic cake but do not know how to bake a chocolate cake). Processes here can be conscious but do not need to be**.** The user uses rules that worked in an earlier, other (mostly likewise) setting in a current setting. He or she uses the roles correctly but they do not work. The goal or plan as defined by the user is incorrect.

3) **Skill based behaviour and mistakes:** This level of behaviour has the lowest level of regulation, as a lot of processes here are automated and can thus be performed without conscious attention. Regulation here cannot change action programs, at best only stop the performance coming from it. Mistakes do occur here when the situation is familiar to the user. The intention or plan is then correct but the execution is not.

The work of Reason can also be compared to that of Zapf et al.(1992). In their work (based on the German Action Theory), Zapf et al start with the comparison with the three levels of action regulation as proposed by Rasmussen:

1) **Intellectual level:** Comparable with knowledge based level by Rasmussen where conscious processing occurs almost all the time.
2) **Flexible action patterns**: Comparable to the rule based level as proposed by Rasmussen where conscious processing happens in schemata but is not needed all the time.
3) **Sensorimotor level:** Comparable to the skill-based level from Rasmussen where processes are almost automated and barely need conscious processing

Theory by Zapf et al.

Around the same time, Zapf et al. made a model similar to the one by Rasmussen, but extended it by adding a **knowledge base for regulation** which is used for developing plans and goals in the first place. This base consists of a) knowledge of facts, b) knowledge of procedures and c) understanding in the sense of mental models.

When comparing the work from Zapf et al with that from Reason and Rasmussen, as discussed previously, two things are noticeable. First of all, Zapf et al add a third dimension. Next to planning and monitoring problems, they add a category for usability problems based on feedback. Second, while Reason does not imply that slips and lapses (which basically differ in regulation level) can also occur during planning (only during execution), Zapf et al combine the dimensions of both regulation level and planning level. This creates a possibility to define type of error by two dimensions:

This base creates a possibility to define type of usability problems by two dimensions:

1) Where in the process did the usability problem occur?

a) Planning
b) Monitoring
c) Feedback

2) What is the level of regulation for this usability problem (as defined by Rasmussen)?

a) Knowledge level
b) Rules level
c) Skills level

Combining these dimensions gives, as described by Zapf et al, eight types of problems which are summarized in the table below:

Knowledge base for regulation			
Knowledge errors			
	Goals/Planning	**Monitoring**	**Feedback**
Knowledge level	Thought problems	Memory problems	Judgment problems
Rules level	Habit problems	Omission problems	Recognition problems
Skills level	Sensorimotor problems (slips/lapses)		

As can be seen above, sensorimotor mistakes happen only at the skills level of behaviour but in all three phases of the process. Mistakes in this category still need to be divided in slips and lapses by questioning for each usability problem found whether the point of execution was reached or not. Next is a short explanation for each type of usability problem or mistake from the table given above:

- **Knowledge problems:** The user cannot make a correct plan for execution because he or she does not know all the (sub) parts or commando’s from the system that is being used. These problems can occur because the instructions about the program or task are inadequate, and can be traced back to the knowledge base for regulation.

Errors that occur in the knowledge level of regulation mostly are complex as there are a multitude of errors possible:

- **Thought problems:** As can be seen, these problems occur while setting up a goal or preparing a planning. While the user knows all the parts of the system (albeit in a very conscious way of processing), the plan or goal that is set up beforehand is incorrect.
- **Memory problems:** Happen during the task monitoring. The plan or goal is correctly set up, but while working with the system the user forgets part of the plan and thus forgets to execute this what either leads to a) possible execution of a task while forgetting a part or b) execution not possible because the part of the plan that was forgotten was necessary for execution.
- **Judgment problems:** Happen during the feedback phase, so after the user has given the system input. The user receives feedback from the system but either does not understand this feedback or interprets this the wrong way.

As explained above, problems on the rules level happen when the actions that are performed are relatively well-known:

- **Habit problems:** Mistakes of these category occur at the beginning of a task. For example, a participant might say that "well, it looks like [something similar] so I figured it will work that way. This type of problem can occur when, for example, a user switch to a new program for an old task or after an interface redesign of a known program.
- **Omission problems:** Happen during monitoring, when a (sub) plan is executed incorrect even when it normally is done. For example, when sending an e-mail one does not click 'send' but goes straight back to the main menu even when this went right three times before.
- **Recognition problems:** Happen during the feedback phase, when feedback provided by the system is misinterpreted, or misunderstood, even when someone did understand it before. It is really important to note that the difference between recognition problems and judgment problems is that judgment problems have to do with newly received feedback while recognition errors have to do with interpreting feedback that has been received (and understood) before.

Last, there is the level of skill-based problems. There is only one category here. As skill-based behaviour is mostly performed at a less conscious level (automated), it is very hard to make a difference in whether the mistake occurred at the planning, monitoring or feedback phase:

- **Sensorimotor problems:** Mistakes where the plan or intention was fully correct but the execution failed. For example, when a participant presses the wrong button but immediately says that this was not his or her intention. The assimilation bias can also be found in sensorimotor problems, as an earlier learned automatism from another situation can lead to the execution of this automatism in the wrong situation. This level of behaviour can be divided into slips and lapses afterwards, depending on the outcome of the action:

  a) **Slip:** When execution goes right after some time (e.g. correcting a spelling mistake during typing)**.**

  b) **Lapse:** Execution end up in incorrect action (e.g. when a participant accidentally goes back to the main screen and knows this is incorrect but is not able to get back to the working screen).

Now that the theory behind usability problem classification has been explained thoroughly, it is time to define what should be done with the data before usability problems can actually be classified into one of the above categories

Prerequisites for data

Most usability tests give a lot of data to work with. Here are some ideas to get started:

1) These guidelines have been developed and tested to classify usability problems. Even though these guidelines might be able to classify individual errors, we strongly advise you to sort out your data by classifying and creating usability problems beforehand.

   Usability problems are created by grouping individual errors, or incidents, together to get a more general description or underlying idea of what went wrong.

   There are multiple methods available for doing getting these Usability problems. For our research, one of the methods as described by Lavery, Cockton and Atkinson (1997) was chosen. Here, similarities are found between individual errors based on multiple aspects of an incident. This is just one method and there are many more (for a comparison see, for example, Hornbæk & Frøkjær, 2008). If you choose another method, make sure that there is still enough details saved from the incidents to be able to classify the problems in the correct way.

2) You should pay attention to the following things (if applicable) when grouping incidents together (because this will make it easier to classify later on in the process):
    a. Intention: Did the participant had a plan for execution or not?
    b. Comparison: Did the participant compare the task with something familiar? (e.g. "oh, this looks just like my old computer, so I should probably do this...")
    c. Feedback: Was the participant able to know what the feedback meant? Has it appeared earlier to him or her during usability testing?
    d. Reappearance: Did the participant made the same mistake before?
3) Before testing, it might be a good idea to measure previous experience as well. As you will read later on, certain types of mistakes also depend on previous experience with a (likewise) system or device.

Following this, you should end up with a list of carefully described usability problems consisting of a collection of similar incidents.

Step-by-step Guidelines

Now that you have your list if usability problems, we will describe the guideline for classifying each problem into a category. Rather similar to the theory described above, distinction will first of all be made between the knowledge mistake and the rest of the schedule, as the knowledge problem in the taxonomy is placed separately. Afterwards, the seven problems of action theory that are left will be first broken down in the three different regulation levels (Dimension 1) which will in turn be broken down to the specific problem categories using the definition of the steps in the action process below (Dimension 2). For each usability problem, follow the steps below.

> ***Note 1:*** *If you are using a computer, tablet or mobile device it is possible to use the hyperlinks in the text to follow the steps in the guidelines. These hyperlinks look like this:* **EXAMPLE.**
>
> ***Note 2:*** *If you feel you aren't able to classify a problem by following the steps, please read the problem classification with examples and control statements at the end of the document to compare your problem and find the best match (***Problem Categories*****).***

Start the steps

Step 1

In this step, a check will be made whether a usability problem is one in the range of knowledge base regulation (a knowledge problem) or not

Relevant questions:

- Did the user miss any knowledge about the buttons, functions, etc. making it impossible to complete the action successfully? *(Yes: Choice 2; No: Choice 1)*
- Did the user receive adequate instruction*? (Yes: Choice 1; No: Choice 2)*

Choices:

1. **Action regulation:** The user received an adequate instruction and had enough knowledge to possibly successfully perform the action: continue to **Step 2.**
2. **Knowledge base for regulation:** The user didn't receive (part of) an instruction or didn't know about certain buttons, functions, etc. It was impossible for the user to complete the action successfully: continue to **Knowledge problems**.

Dimension One (Regulation level):

Step 2

If the usability problem is not a knowledge problem, the next step is to find out in which level of regulation it occurred: either knowledge-based, rule-based or skill-based.

Relevant questions:

- Did the outcome comply with the intention of the user? (Even if the outcome wasn’t successful/useful for the task?) *(yes; choice 2, no; choice 1)*
- Does the user exclaim out loud that this wasn’t what he meant to happen? *(yes; choice 1, no; choice 2)*
- Did the user accidentally press the wrong button/link or next to a button/link? *(yes; choice 1, no; choice 2)*

Choices:

1. **Skill level:** The user performed this action with little conscious thinking or almost automatically. The user seemed familiar with the situation. His or her plan of action was correct, even though the execution was not necessarily: continue to **Sensorimotor problems.**
2. **Rules level or Knowledge level:** The user used quite a lot of conscious control for this action, the situation was rather unknown or new to him or her, and most likely there was an incorrect plan of action: continue to **Step 3.**

Step 3

If your problem was not a skill-based, sensorimotorical one, then it is either a knowledge-based or a rule-based problem

Relevant questions:

- Did the user use an (implicit) if/then statement or rule (if I do this.....then this will happen) in his plan for the action? *(yes; choice 1, no; choice 2)*
- Did the user encounter this same problem before in the same manner? *(yes; choice 1, no; choice 2)*
- Did the user find that this situation resembled something that he knew from another situation or recognise the situation? *(yes; choice 1, no; choice 2)*

- Did the user need to form a new plan for this action? *(yes; choice 2, no; choice 1)*
- Does the user say that he is going to try something new (but there is an intention/plan)? *(yes; choice 1, no; choice 2)*

Note: If there is no intention or plan, it can never lead to a problem. So there has to be a plan or intention!

Choices:

1. **Rules level:** The user performed the action on the Rules level. There was a (schematic) plan that was probably based on earlier experiences with a (likewise) system, but this planning was incorrect, leading to an incorrect execution (in most cases): continue to **Step 4.**
2. **Knowledge level:** The user performed the action on the Knowledge level. The user made a new plan, which was executed step-by-step. There might have been an information overload, as the user did get an introduction to the system but this might be a lot of information at once: continue to **Step 6.**

Dimension Two (steps in action process):

Rule-based

Step 4

Your usability problem is rule-based. The next question is whether it took place during the planning, monitoring or during the feedback phase.

Relevant questions:

- Was the plan that the user formed adequate? *(yes; choice 2, no; choice 1)*
- Did the problem occur before the execution of the action? *(yes; choice 1, no; choice 2)*
- Was the action based on a habit of the user (from another situation)? *(yes; choice 1, no; choice 2)*
- Did a feature of the application lead to a wrong assumption/plan? *(yes; choice 1, no; choice 2)*

Choices:

1. **Planning:** The usability problem occurred in the phase of planning, so before the action was executed: continue to **Habit problems.**
2. **Monitoring or Feedback:** The plan for execution was right, but the execution went wrong or feedback interpretation or usage after action performance went wrong: continue to **Step 5.**

Step 5

Your usability problem is rule-based and took place either during monitoring of during the feedback phase. This step is to find out when:

Relevant questions:

Monitoring:

- Did the user forget to execute a part of the plan? *(yes; choice 1, no; choice 2)*
- Did the error occur during a sub action? *(yes; choice 1, no; choice 2)*
- Was this part of the plan well known? *(yes; choice 1, no; choice 2)*

Feedback:

- Did the user complete the task? *(yes; choice 2, no; choice 1)*
- Did the user have trouble understanding or interpreting feedback by the program? *(yes; choice 2, no; choice 1*)
- Was this known/earlier encountered feedback? *(yes; choice 2, no; choice 1)*
- Was there a lack of feedback that confused the user? *(yes; choice 2, no; choice 1)*
  a. *(If/then construction: if I finish, then there will follow feedback. If this doesn't follow this is an Recognition problem)*
- Was there feedback present that the user didn't see which led to a problem? *(yes; choice 2, no; choice 1)*

Choices:

1. **Monitoring:** The problem encountered took place during execution of a (sub) plan and not afterwards. Therefore, it is an omission problem: continue to **Omission problems.**

2. **Feedback:** The problem encountered took place after the execution of a (sub) action. It either happened because feedback was misinterpreted or because there was a lack of feedback. It is a recognition problem: continue to **Recognition problems.**

Knowledge-based

Step 6

Your usability problem is knowledge-based. The next question is whether it took place during the planning, monitoring or during the feedback phase.

Relevant questions:

- Was the plan that the user formed adequate? *(yes; choice 2, no; choice 1)*
- Did the problem occur before the execution of the action? *(yes; choice 1, no; choice 2)*
- Did a feature of the application lead to a wrong assumption/plan? *(yes; choice 1, no; choice 2)*

Choices:

1. **Planning:** The usability problem occurred in the phase of planning, so before the action was executed. It is a thought problem: continue to **Thought problems.**
2. **Monitoring or Feedback:** The plan for execution was right, but the execution went wrong or feedback interpretation or usage after action performance went wrong: continue to **Step 7.**

Step 7

Your usability problem is a knowledge-based problem that had a good action plan. It did went wrong either during monitoring or feedback. This step is to check at which point it went wrong. Relevant questions:

Monitoring

- Did the user forget to execute a part of the plan? *(yes; choice 1, no; choice 2)*
- Did the error occur during a sub action? *(yes; choice 1, no; choice 2)*

Feedback

- Did the user complete the task? *(yes; choice 2, no; choice 1)*
- Did the user have trouble understanding or interpreting feedback by the program? *(yes; choice 2, no; choice 1)*
- Was this new/unknown feedback? *(yes; choice 2, no; choice 1)*

Choices:

1. **Monitoring:** The problem encountered took place during execution of a (sub) plan and not afterwards. Therefore, it is a memory problem: continue to **Memory problems.**
2. **Feedback:** The problem encountered took place after the execution of a (sub) action. It either happened because feedback was. It is a judgment problem: continue to **Judgment problems.**

Problem Categories

Below each category description you will find a short set of control statements and examples. Use these to make sure your usability problem belongs to the right category in case of doubt.

Knowledge problems

The user cannot make a correct plan for execution because he or she does not know all the (sub) parts or commando's from the system that's being used. These problems can occur because the instructions about the program or task are inadequate, and can be traced back to the knowledge base for regulation.

- *The user has not performed the task with the tested device before*
- *The user has not worked with a (very) similar device before*
- *The user states that he or she has no idea how to do this, since it is unlike anything witnessed before*
- *Possible to check by questionnaires about previous experience with the device tested, or similar devices.*

- *The user didn't receive the correct instructions about buttons, touchscreen or functions beforehand to perform the task*

GO BACK TO STEP 1

Thought problems

As can be seen, these problems occur while setting up a goal or preparing a planning. While the user knows all the parts of the system (albeit in a very conscious way of processing), the plan or goal that is set up beforehand is incorrect.

- *The user received adequate instructions.*
- *The user shows an incorrect plan of action when thinking out loud.*
- *The user wants to try something to see if it will work.*

GO BACK TO STEP 6

Memory problems

Happen during the task monitoring. The plan or goal is correctly set up, but while working with the system the user forgets part of the plan and thus forgets to execute this what either leads to a) possible execution of a task while forgetting a part or b) execution not possible because the part of the plan that was forgotten was necessary for execution.

- *The user has a correct plan of action before actually doing something but forgets to execute a part of it.*
- *The plan that the user wants to execute is newly formed/no prior experience with the plan.*
- *It is only a memory problem if the user forgot to perform the action. If he tried to perform it but he failed this is another type of problem (For example: when a user is trying to click the save button, but it doesn't react and the user doesn't know what is happening, this would qualify as a judgment problem).*
- *If the user tried to click it but failed in the action and doesn't notice it, this is a sensorimotor problem and NOT a memory problem.*

GO BACK TO STEP 7

Judgment problems

Happen during the feedback phase, so after the user has given the system input. The user receives feedback from the system but either does not understand this feedback or interprets this the wrong way.

- *The user notices the feedback (either by responding to it verbally or behaviourally) but does not know what to do with it, or act wrong on it.*
- *The user indicates to not understand this feedback ("Huh? What is this about?" or something likewise).*
- *The user did not receive this feedback before, and if he or she did receive it, not understand it then either.*

GO BACK TO STEP 7

Habit problems

Mistakes of this category occur at the beginning of a task. Participants want to perform an action or plan that in itself is not wrong but the moment of using this action is wrong. For example, a participant might say that "well, it looks like [something similar] so I figured it will work that way. This type of problem can occur when, for example, a user switch to a new program for an old task or after an interface redesign of a known program.

- *The user is familiar with the system or task, or a likewise system or task.*
- *The action the user performs in itself is not wrong. The action could have been correct in another situation. The place or situation is wrong.*
- *The user exclaims this will probably work like a similar situation he knows, or that he wants to try if this is the same as another situation.*

GO BACK TO STEP 4

Omission problems

Happen during monitoring, when a (sub) plan is executed incorrect even when it normally is done. For example, when sending an e-mail one does not click 'send' but goes straight back to the main menu even when this went right three times before.

- *The user already talks or thinks out loud about the next step that has to be performed (e.g. a user sending an e-mail thinking out loud: "I will have* *to go to outbox to check whether I have sent it" who consequently forgets to hit the send button and goes straight to outbox, only to discover that the mail was not send).*

- *The user has performed the task correctly before.*
- *The plan was adequate for the task.*

GO BACK TO STEP 5

Recognition problems

Happen during the feedback phase, when feedback provided by the system is misinterpreted, or misunderstood, even when someone did understand it before. It is really important to note that the difference between recognition problems and judgment problems is that judgment problems have to do with newly received feedback while recognition errors have to do with interpreting feedback that has been received (and understood) before.

- *The user shows no indication of noticing a feedback message from the system halfway a task (or during a subtask) when it appears. The user continues without the feedback.*
- *Feedback is present but the user didn't notice it, due to the feeling that he was already finished and didn't need to pay attention anymore.*
- *The user has shown intention of noticing this feedback message earlier.*
- *The user has shown before to know the meaning of this feedback message, either in this system or a likewise system.*
- *The user indicates that he or she is missing feedback: either by indicating directly ("it would have been nice if the system would tell me what to do next") or indirectly ("I don't know what to do next..?").*
- *If the user doesn't notice a lack of feedback due to automatized behaviour (for example: trying to check a box and click on continue, but the box is still empty and the page doesn't react) it is a sensorimotor problem if the user understands what went wrong and a recognition problem if he doesn't understand the feedback.*

GO BACK TO STEP 5

Sensorimotor problems

Mistakes where the plan or intention was fully correct but the execution failed. For example, when a participant presses the wrong button but immediately says that this was not his or her intention. The assimilation bias can also be found in sensorimotor problems, as an earlier learned automatism from another

- *The user states that he or she knows what to do, or describes a (correct!) plan of action.*
- *The user immediately indicates that the thing that went wrong was a mistake, or even explains what he or she intended to do instead (note: this description must be correct!)*
- *The error is a physical one: for example, knowing what the next step is but accidentally pressing a wrong button because they are too close*
- *As there is no separate feedback level for the sensorimotor level, it is possible that a user tries to click a button, misses, and doesn't notice this due to automatized behaviour. This also qualifies as a sensorimotor problem.*

GO BACK TO STEP 2

Appendix C: Rules for binary coding

Five algorithmic-like rules were created to help to assign the binary codes to the right groups. These five rules are described below:

Rule #1:

```
If
    The first number is ‘0’
And
    The second number is ‘0’
And
    The third number is ‘0’
Then
    Return: ‘Zero’
```

Rule #2:

```
If
    The first number is ‘0’
And
    The second number is ‘1’
And
    The third number is ‘0’
Then
    Return: ‘Group 2’
```

Rule #3:

```
If
    The first number is ‘0’
And
    The second number is ‘1’
Then
    Return: ‘Group 2’
And
    If
    The third number is ‘0’
Then
        Return: ‘Group 1’
Or
If
    The third number is ‘1’
Then
        Return: ‘Group 3’
```

Rule #4:

```
If
    The first number is ‘1’
And
    The third number is ‘0’
Then
    Return: ‘Group 1’
```

Rule #5:

```
If
    The first number is ‘1’
And
    The third number is ‘1’
Then
    Return: ‘Group 3’
```

Appendix D: Additional analyses and results

A number of analyses were performed in the MOBILE.OLD project that were not of direct interest to the persistency patterns of problems. These additional analyses are discussed shortly here, to give an idea of the other results that were retrieved for the MOBILE.OLD project. This section will feature the discovery rate for user errors and usability problems, the analysis of the time on task and the analysis of the ASQ scores for user satisfaction.

Discovery rates

The study investigated the usability of the applications that were designed in the MOBILE.OLD project by user testing over multiple trials. Usability testing was performed to see what usability problems were present. An extended matching protocol was used, which includes a step that uses errors classifications to form user errors. The discovery rates were investigated, to see if the extended matching protocol was able to identify a large percentage of the usability problems. In total, 148 unique user errors were found to be caused by 49 unique usability problems. The number of user errors and usability problems found of all different users can be seen in table D1. These are the problems that were found in total over three trials.

Table D1

Number of errors and problems discovered by users

User	Number of User errors	Number of Usability problems
01	56	28
02	49	30
03	23	15
04	51	26
05	39	25
06	34	23
08	40	26
09	40	23
10	38	21
11	38	20
12	56	32
13	48	30
14	35	19
15	45	23
16	31	18
17	31	20
18	54	32
19	19	16
20	24	15
Total unique discoveries	148	49

By looking at how many different users discovered an error or problem, it is possible to make a prediction about how many errors or problems have not yet been discovered. A prediction about how many errors or problems has not been discovered yet can be found in figure D1a and b. The intercept with the y-axis predicts the number of problems or errors that zero users found, making it possible to predict the total number of problems or errors. These figures predicted that in total approximately 75 user errors and 6 usability problems were not yet discovered, which are respectively 66% of all user errors and 89% of all usability problems.

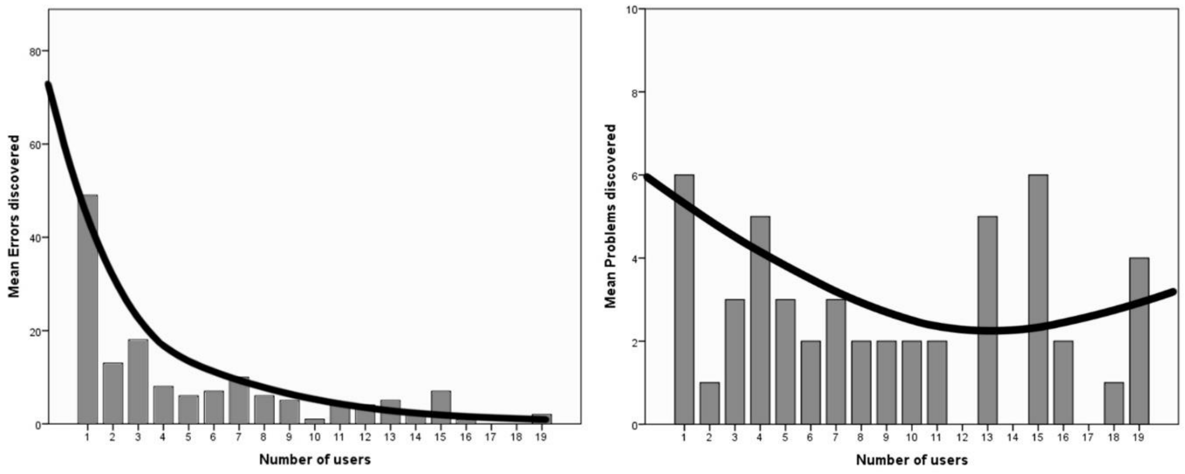

Figure D1a & b. Number of times that a user error (a) and usability problem (b) were discovered.

These results show that the extended matching protocol was able to discover a high percentage of usability problems and can therefore be seen as a success. The percentage for user errors is considerably lower at 66% than the percentage for the problems, which reflects how much variety and individual differences can be seen in behaviour, even though these behaviours are caused by the same problems in usability.

It may not be a problem that the percentage of found user errors is lower, as they are expected not to be of interest to the developers of the applications. Developers are interested in what they actually need to improve in their design, so they are expected to just want to receive the usability problems. Improving the discovery rate for user errors would take a lot of effort to get a small improvement in the discovery rate for the usability problems, which is already very good at 89%. As a comparison, Nielsen (1994) stated that that 75% should be enough in a usability study. In conclusion, the use of user errors helps to achieve a high discovery rate for usability problems. The discovery rate of user errors is less relevant, as they do not need to be delivered to developers. In other type of studies, such as behavioural studies, the user errors may prove to be more interesting to investigate.

References

Nielsen, J. (1994). Estimating the number of subjects needed for a thinking aloud test. *International Journal of Human-Computer Studies*, *41*(3), 385–397. doi:10.1006/ijhc.1994.1065

Time on task

The MOBILE.OLD project was interested in learning if the elderly users were improving over time using the applications. In other words, were the elderly learning? A performance measure that was used to investigate this was the time that users spent on a task. Two hypotheses were formed to investigate the progress of time on task:

H1. Users will become faster in performing a task over the tree trials.

H2. A higher level of previous experience will lead to a lower time on task.

Generalized estimated equations were used to investigate these hypotheses. No violations of the assumptions for this analysis were found (Zuur et al., 2010), as can be seen in Appendix F. The dependent variable time on task was measured in seconds, with possible values from zero to positive infinite, so a gamma distribution was chosen for the model. The results of the generalized estimated equations can be found in table D2. To account for learning effects and effects of fatigue, an autoregressive working correlation matrix was chosen for the generalized estimated equations. The syntax for the performed analysis can be found in Appendix G.

Table D2
Results from generalized estimated equations for time on task

			95% Confidence Interval			
Parameter	Beta	Standard Error	Lower bound	Upper bound	Significance	QICC
Intercept	8.749	2.021	4.787	12.711	0.000	250.719
Trial 1	0[a]					
Trial 2	-0.553	0.107	-0.763	-0.342	0.000	
Trial 3	-0.316	0.135	-0.580	-0.052	0.019	
Previous experience	-0.169	0.098	-0.362	0.023	0.085	

[a] parameter has been set at 0, because it is redundant. Other parameters are compared to this one.

This table shows that the users were indeed faster in trial 2 and 3, as compared to trial 1. The negative beta values and confidence intervals show that a negative relation between trial and time on task can be seen as very certain. Exponentiation of the beta values showed that users were 42% faster in trial 2 and 27% faster in trial 3 compared to trial 1. The beta value for previous experience was in accordance with the hypothesis H2, as it showed a negative value, but the confidence intervals showed that there was no certainty for the direction of the effect. The effect approached significance, but did not show enough certainty to assume it as true.

The found results for time on task over the multiple trials are in accordance with the hypothesis H1, which will therefore be accepted. Hypothesis H2 will be rejected, as the results pointed towards an effect for previous experience but could not confirm it. The results are in accordance with the earlier discussed findings by Westerman, Davies, Glendon, Stammers and Matthews (1995), as the results show that the elderly are improving significantly over time. The elderly are clearly capable of learning and improving in the use of mobile devices. In trial 3, a week after the other two trials, the elderly were a little bit slower than in trial 2, but still much better than in trial 1. This means that there was a decrease in performance during the time between trial 2 and 3, but as the elderly users still performed better in trial 3 than in trial 1, they did acquire a certain level of expertise. It may be interesting to investigate this decrease further and to compare this with younger users to see the learning effects over a larger period than a week.

In relation to problem severity and persistence, time on task may be used to detect problems that keep hurting user performance over a number of trials. Quick improvements of time on task would indicate a low level of persistency, while a consistent time on task could indicate a persistent problem. It is important to have a reference category for the time on task, to be able to determine if a consistent time on task means that the task was performed consistently fast or consistently slow. This could for instance be done by combining the time on task measurement with measuring the deviation from the optimal solution (Hornbæk, 2006), which could be used as a heuristic to determine if problems need to be investigated further. Another option would be to compare the consistent time on task with the time on task by a system expert. This method of detecting persistency lacks a way to look at specific problems, as all problems made in a task influence the time on task. The method would be most suited as a heuristic for detecting problem severity, rather than the focal point of a usability evaluation.

References

Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, *64*(2), 79–102. doi:10.1016/j.ijhcs.2005.06.002

Westerman, S. J., Davies, D. R., Glendon, A. I., Stammers, R. B., & Matthews, G. (1995). Age and cognitive ability as predictors of computerized information retrieval. *Behaviour & Information Technology*, *14*(5), 313–326. doi:10.1080/01449299508914650

ASQ scores

The MOBILE.OLD was not only interested in looking at how well the elderly users were able to interact with their applications, but also wanted to know how much the elderly users liked to use the applications. As is apparent from the ISO definition of usability (ISO, 2008), satisfaction is one of the three basic factors for usability and is therefore an important aspect to take into account during usability evaluation. ASQ scores were retrieved to investigate the satisfaction among the elderly users. Three hypotheses were formed about the effects of the ASQ scores:

H1. The average ASQ score will become higher in the later trials than in the earlier trials.

H2. The average ASQ scores of users will become higher if users show a higher task completion.

H3. The ASQ scores of users will become higher if users show a lower time on task.

Descriptive statistics showed that the ASQ scores were relatively high overall (*M*=4.93, *SD*=1.62). The ASQ scores were investigated using autoregressive generalized estimated equations to see the effects of satisfaction in general and over three trials. The assumptions for generalized estimated equations were checked (Zuur et al., 2010), but no violations were found. Check can be found in Appendix F. A gamma distribution was chosen, as the values could only be real values and positive. The results of the analysis can be seen in table D3.

Table D3
Results from generalized estimated equations for ASQ score

			95% Confidence Interval			
Parameter	Beta	Standard error	Lower bound	Upper bound	Significance	QICC
Intercept	-3.682	2.142	-7.881	0.516	0.086	162.458
Trial 1	0[a]					
Trial 2	0.105	0.049	0.010	0.200	0.030	
Trial 3	0.111	0.052	0.010	0.213	0.032	
Task completion (a lot of help)	0[a]					
Task completion (some help)	-0.049	0.079	-0.203	0.106	0.537	
Task completion (without help)	-0.034	0.114	-0.257	0.190	0.768	
Time on task (s)	-0.001	0.001	-0.002	0.000	0.067	
Age	0.073	0.030	0.015	0.131	0.014	

[a] parameter has been set at 0, because it is redundant. Other parameters are compared to this one.

The results showed that there was an effect for the trials on the ASQ score. Trial 2 and 3 showed a higher average ASQ score than trial 1. Exponentiation showed that users scored 11% higher in trial 2 and 11% higher in trial 3 compared to trial 1. The positive beta values and small confidence intervals showed that this effect can be seen with a lot of certainty. An unexpected significant effect was found for age on the average ASQ scores. A higher age was found to lead to a higher level of satisfaction. Exponentiation showed that an increase of age by one unit, increased the ASQ score by 8%. Task completion, time on task and geekism all showed no significant effects.

The elderly users were generally satisfied with the applications, with a mean score for ASQ of 4,296 on a scale between 1 and 6. The results also showed that the elderly became more satisfied when performing the tasks multiple times. The ASQ score in trial 2 and 3 were higher than trial 1, providing proof in favour of hypothesis H1 and is therefore accepted. No effects were found for time on task and task completion, so the hypotheses H2 and H3 are rejected. The unexpected effect for age, showing that older users were more satisfied than younger users, may be caused by the expectation level that the elderly showed. The 'older' elderly users may have had lower expectations about their own skill and were positively surprised about their interactions, while the 'younger' elderly users had a little more critical opinion and expected more from the system and their own skills.

The ASQ scores may prove to show potential as a subjective measure for problem severity in the same heuristic-like manner as was described for the time on task. If users remain unsatisfied about a certain task, it may be that a function is too difficult or not logical for a user. Even though users may not encounter problems in a function, users can still be very unhappy about using it. Nielsen & Levy (1994) showed that there is a strong positive association between preferences and performance in usability, but also found that there are a lot of examples of preferred designs that do not show the best performance. The preferences of users could also be taken into account when prioritising the improvements on a product, as they may not want the objectively best design. These findings seems to reflect that problem severity, which is mostly aimed at performance measures, could be extended with a satisfaction measure. The satisfaction that users experience in using a product and the impact on severity is mentioned by Nielsen (1995), but is not included in the definition of severity. Nielsen calls this factor *market impact*, an aspect of a product that influences its popularity due to positive or negative opinions of the audience. Future research can consider the addition of market impact or another form of satisfaction to the equation that defines problem severity. This would mean that usability research should also try to detect design issues that impact the subjective user experience, rather than their objective performance.

References

ISO (2008). *Ergonomics of human-system interaction — Part 171: Guidance on software accessibility (ISO 9241-171).* London: International Standards Organization.

Nielsen, J. (1995, January). Severity Ratings for Usability Problems. Retrieved January 12, 2015, from http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

Nielsen, J., & Levy, J. (1994). Measuring usability: preference vs. performance. *Communications of the ACM*. doi:10.1145/175276.175282

Appendix E: List of usability problems

UP	Description	UP	Description
1	Instruction on editing words not adequate	26	Text error message is unclear
2	Not clear when you can and cannot drag the screen	27	Suggestions are unclear (how to use these)
3	Buttons are unclear (icons/words)	28	Page isn't clear enough to show what it does
4	Help button isn’t finished	29	Feedback given by application is not clear (loading/checkbox/button/lining around box)
5	Translations are wrong on screen	30	Program needs a second time to load
6	12 hour clock (AM/PM) is not clear	31	Tiles are an unclear concept
7	Instruction on keyboard not adequate	32	Unclear that something is already selected/open when user starts
8	Cursor is unclear (deleting a word, where does it start)	33	Thinks that two screens are the same, even though they are different
9	Scroll lists looks too much like room around it? (clicking in the list more obvious)	34	Unclear that something is a button
10	Lack of feedback	35	Instruction on touchscreen not adequate
11	Title looks too much like a button	36	App purpose is unclear
12	Lack of instruction on screen about rule or steps	37	Scrollbar on the right side is confusing
13	Instruction on scrolling/dragging not adequate	38	Unclear when you are able to scroll
14	Photo function isn't directly available	39	Unclear that GPS is not used at the moment
15	Clearer distinction between buttons (On board vs. App buttons)	40	Navigation structure is unclear
16	‘Instellingen’ (settings) too prominent	41	There is too little room between the buttons
17	Text looks too much like button	42	Scrolling is too responsive, needs a delay
18	Names are very unclear for (navigation) functions	43	Buttons aren’t responsive enough
19	Not able to send route after viewing it	44	Buttons react to quickly, need delay for scrolling to occur
20	Unclear that same button can be used multiple times	45	Shape of the remote is wrong, too big. Not able to press button properly
21	Text is too unclear or small, instruction not readable	46	Buttons next to keyboard still react when you try to click them/beside keyboard
22	Rearrangement necessary for buttons / dropdown menu	47	Name of function is unclear
23	After giving points video screen doesn't come back for adding	48	Button is too small, isn't spotted
24	Contrast isn't strong enough for background during popups	49	Better instruction on remote required (how to use in combination with Apps/TV)
25	No consistent way of working in apps and over apps		

Appendix F: Checking of assumptions for GEE

To check if the assumptions that are required to use the generalized estimated equations, the protocol by Zuur et al. (2010) was used. The assumptions for all analyses that were done will be discussed one by one. The syntax for all the analyses can be found in appendix G.

Incident frequency

The complete range of values for the incident frequency was used, because no impossible values were found. To view the distribution of the incident frequency, boxplots were made for the raw residuals. These boxplots can be found in figure F1. The boxplots showed that some of the patterns incorporated outliers in the distribution. These outliers were checked using the video files of the trials and were not found to be caused by any special circumstances. The outliers were therefore not removed from the dataset. It may be possible that these outliers are causing overdispersion. Further inspection of the descriptive statistics indeed showed that the standard deviation was high and that the variance was greater than the mean. The variances of the boxplots for each pattern do not appear to be equally distributed. Therefore, homoscedasticity cannot be assumed. No zeros were found in the dataset.

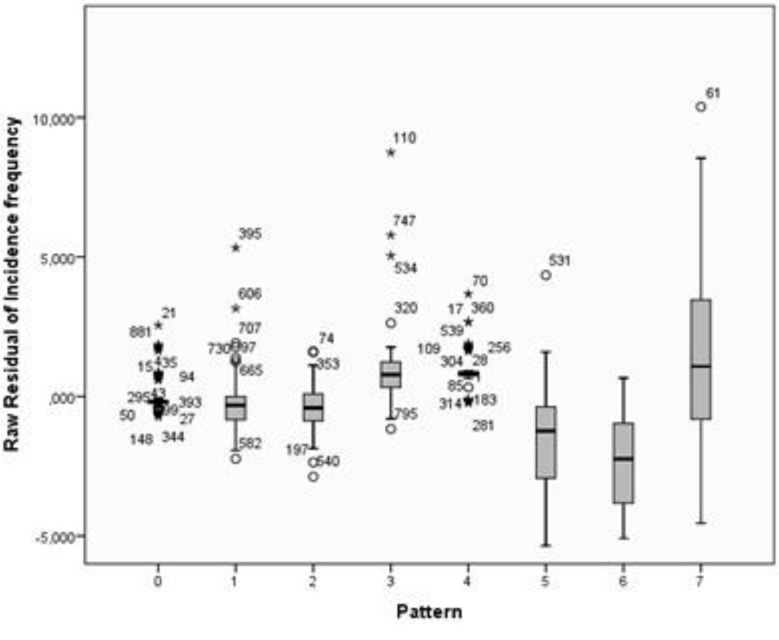

Figure F1. Boxplots for the raw residuals of incident frequency.

A histogram was made for the raw residuals of incident frequency to check if assumptions could be made about the normality of the residuals. This histogram, including a normal curve that has been fitted to the data, can be found in figure F2. The figure shows that the data resembles a normal distribution, so normality will be assumed. Since the dependent variable consists of counted values, a negative binomial distribution was chosen to use in the generalised estimated equations. The negative binomial distribution was chosen in favour of the Poisson distribution, due to the possibility of overdispersion.

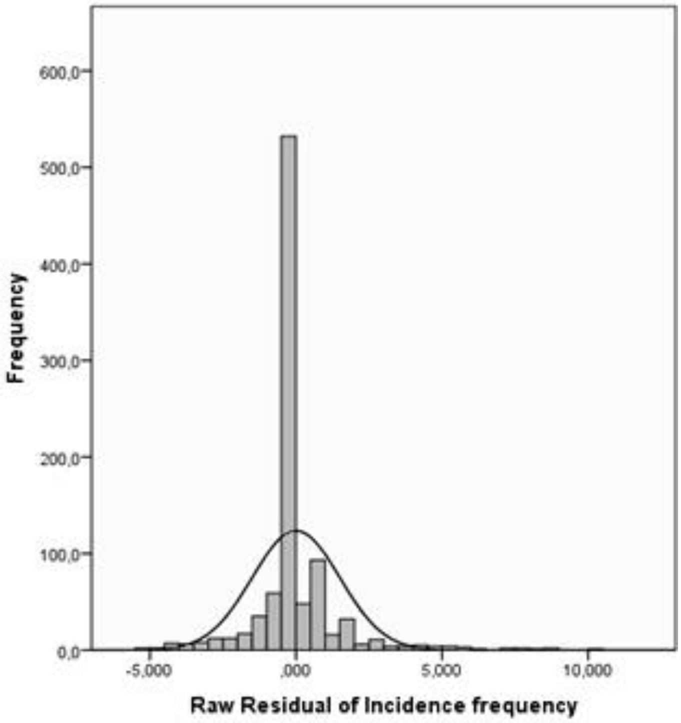

Figure F2. Histogram for the raw residuals of incident frequency to check normality.

Disappear early

The binary data for the presence of a disappear early trend showed no impossible values, therefore the complete range of values was used in the data analysis. Boxplots was made for the raw residuals to investigate the variance and outliers. Separate boxplots for the different patterns were made and can be seen in figure F3.The boxplots showed a couple of outliers, but these were not removed after further investigation. The variance for the different boxplots is equally distributed, showing the presence of homoscedasticity.

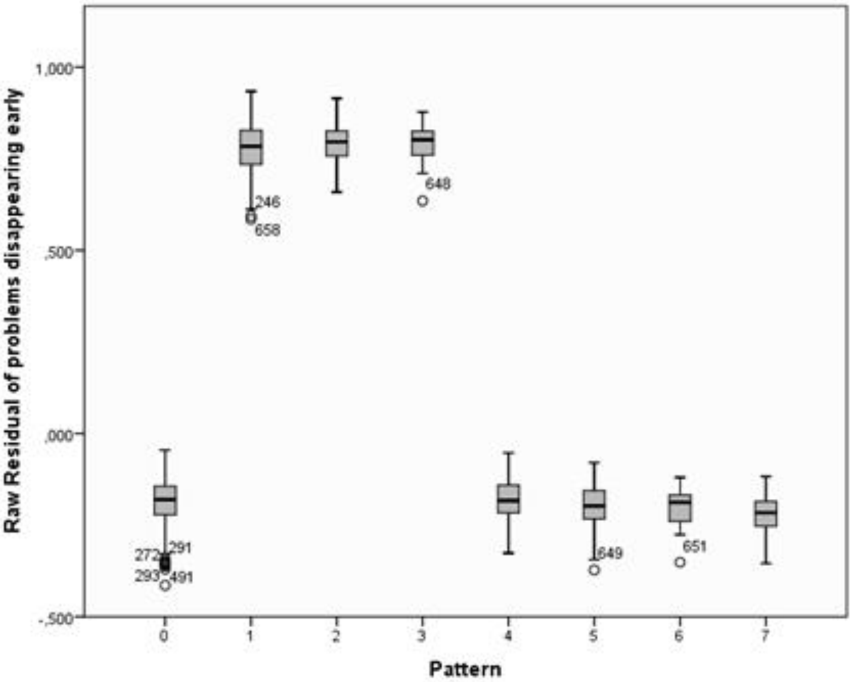

Figure F3. Boxplots for raw residuals of problems disappearing early.

A histogram was made for the raw residuals to investigate the normality of the raw residuals. This histogram can be seen in figure F4.The histogram shows a distribution with two peaks, so it is difficult to apply a fitting normal curve over this data. However, the two peaks seem to be distributed normally on their own. These results do not make it possible to assume a normal distribution for the raw residuals of the dataset.

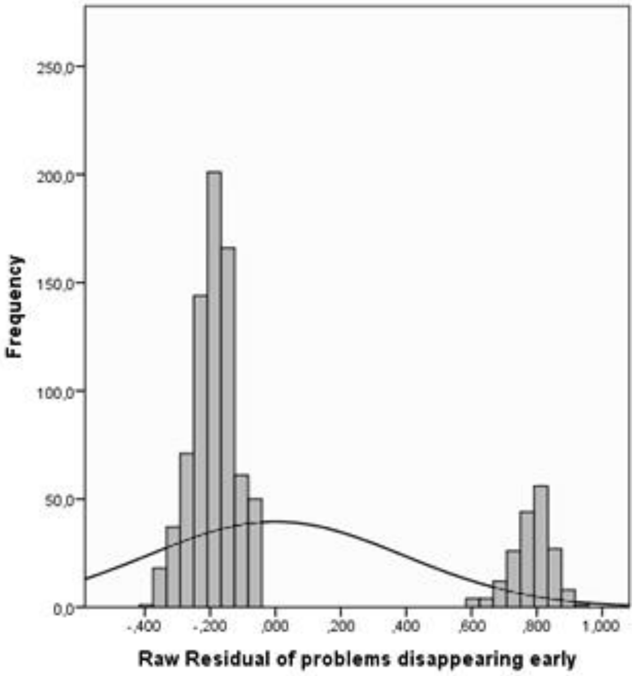

Figure F4. Histogram for raw residuals of problems disappearing early to check normality.

Appear late

The data for the appear late trend also showed no impossible values. Boxplots were used again to visualize the variances and check for outliers in the raw residuals of the appear late trend. Boxplots for the different patterns can be found in figure F5.No clear outliers were present and since no special circumstances were experienced during testing, the outliers were not removed. The variances of the different boxplots are distributed almost equally, so homoscedasticity was assumed.

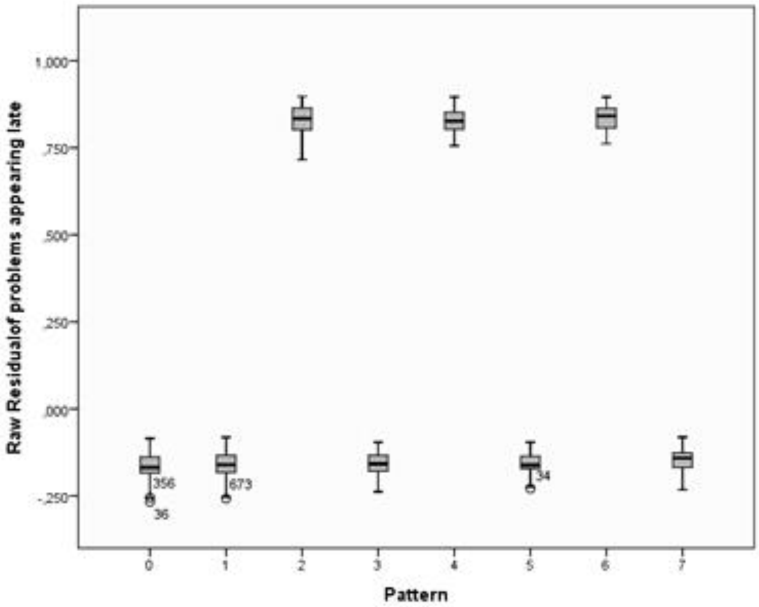

Figure F5. Boxplots for the raw residuals of problems that appear late.

To investigate the normality assumption of the data, a histogram was made for the raw residuals of the appear late trend. This histogram, which can be found in figure F6, showed the same kind of distribution as the histogram for the disappear early pattern. As this figure shows two peaks in frequency, a normal curve cannot be fitted. The normality assumption is therefore violated.

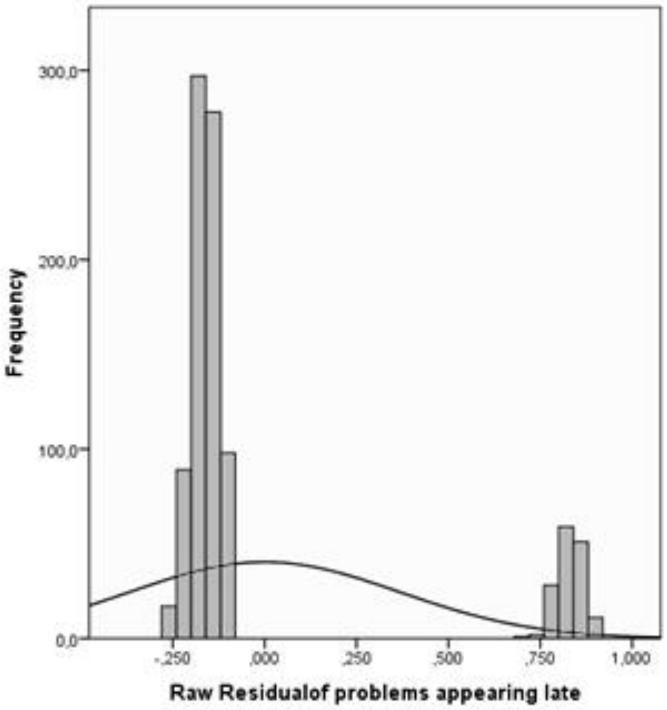

Figure F6. Histogram for the raw residuals of problems appearing late to check normality.

Persistent

The data for the persistent trend did also not show any impossible values. Just like the other checks, first a boxplot, found in figure F7, was made to investigate the variances and check for outliers. The data shows that there are a lot of outliers, mostly for lower values. There were no special circumstances found for these values, so they were kept in the dataset. The variances of the boxplots are not equally distributed, indicating the presence of heteroscedasticity.

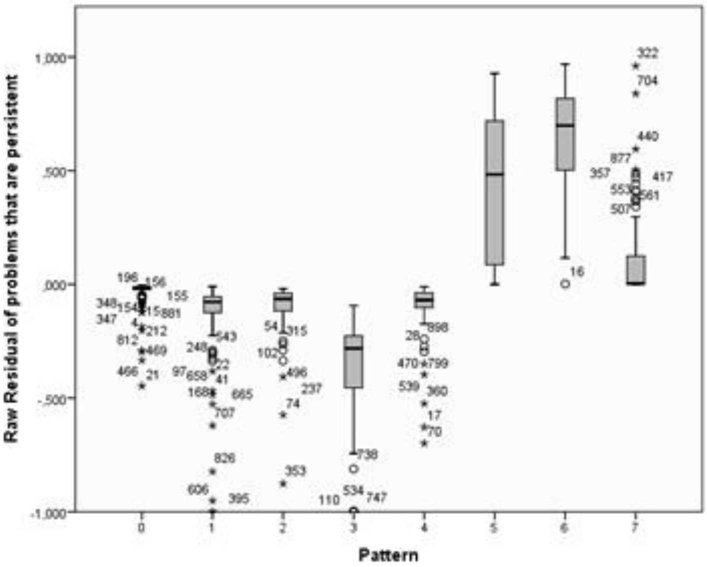

Figure F7. Boxplots for the raw residuals of problems that are persistent.

To investigate the normality of the data, a histogram was made for the raw residuals of the persistent trend. This histogram, as seen in figure F8, shows a distribution that resembles a normal distribution. Normality will therefore be assumed for the persistent trend. As the dependent variable has only binary values, a binomial distribution is advised for the generalized estimated equations.

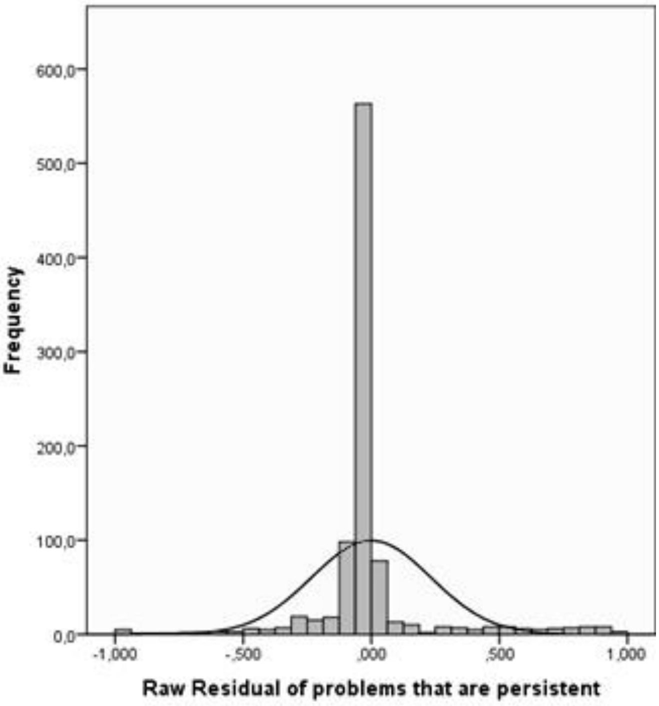

Figure F8. Histogram for the raw residuals of problems that are persistent to check normality.

Average ASQ scores

No impossible values were found which needed to be removed directly, so the complete range of values for the average ASQ were used for further analyses. To check for outliers and to view the variance for the average ASQ score, the raw residuals from the average ASQ scores were used to make a boxplot. This boxplot, with the raw residual on the y-axis and trial on the x-axis, can be seen in figure F9. The boxplot showed a few outliers for the values lower than the average in the second trial. These outliers were checked using the raw data to see if they needed to be removed or not. As there were no special circumstances found, the outliers were kept in the dataset. The variances for the three trials are very close to equally distributed, so homoscedasticity was assumed.

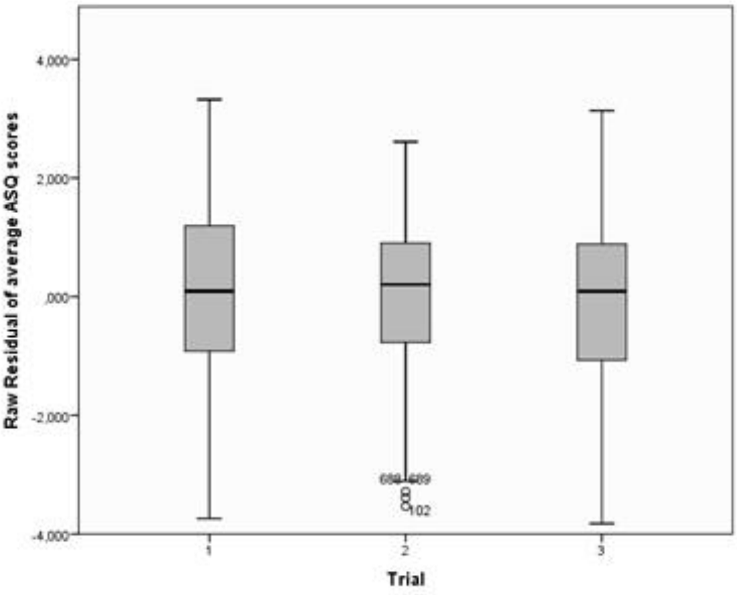

Figure F9. Boxplots for the average ASQ scores over three trials.

A histogram from the raw residuals of average ASQ scores was made to investigate the presence of normality in the data. This histogram, which can be seen in figure F10, showed that the residuals were distributed normally. The data was further investigated using generalized estimated equations. The ASQ scores could only take on positive and real values, so a gamma distribution was chosen.

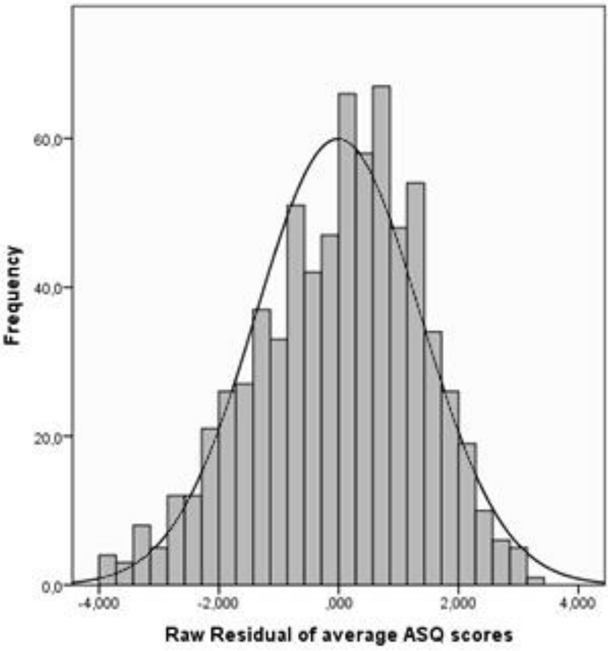

Figure F10. Histogram for the average ASQ scores over three trials.

Time on task

Before checking the assumptions, an impossible value was removed for one user. When looking at the raw data it showed that one user started task 1 during the second trial before the camera was turned on. He thought it was not necessary to wait for instructions, because he thought he already knew what the task he would be asked to perform. After removing the impossible value, boxplots were made showing the raw residual of time on task on the y-axis and the different trials on the x-axis. The boxplots, which were used to check for outliers and to view the variances of the different tasks, can be found in figure F11. The figure shows that there were a lot of outliers for almost every task, indicating that there may be a lot of individual differences between the users for this variable. The raw data was checked for special circumstances, but these were not found. As outliers in time on task can be expected and it is relevant to save the values to reflect individual differences, the outliers were not removed. The variances among the different tasks for time on task were very did not violate the assumption of homoscedasticity.

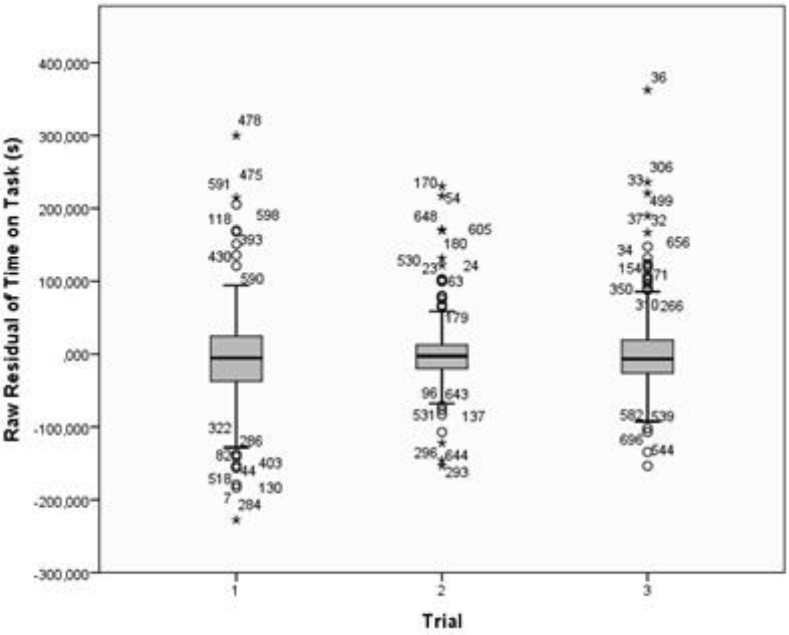

Figure F11. Boxplots for Time on task for each separate trial.

To check the assumption of normality, a histogram was made for the raw residuals of time of task. This histogram can be found in figure F12. The figure shows that the data is nearly normally distributed. As the variable time on task is measured in seconds, it seems probable that a gamma distribution will be most fitting.

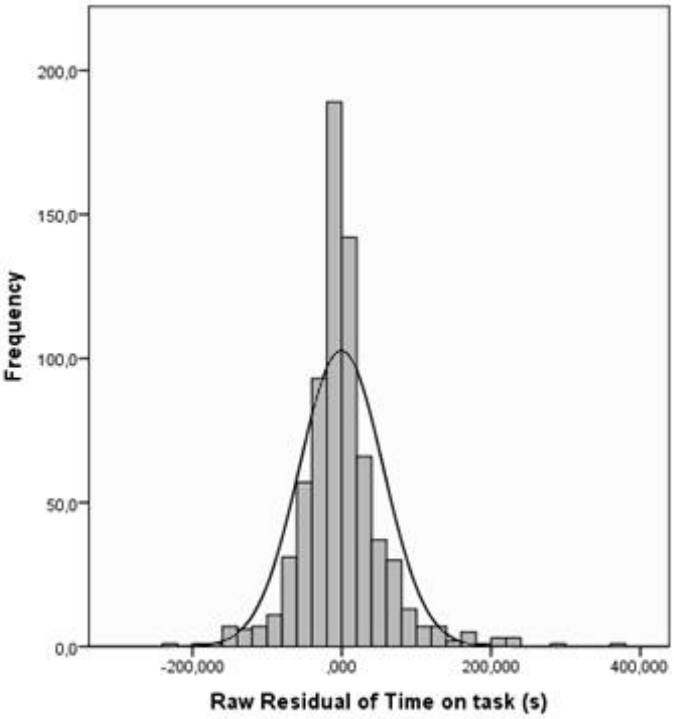

Figure F12. Histogram for raw residuals of time on task scores to check normality.

Appendix G: Syntax

Incident frequency

Generalized estimated equations

```
* Generalized Estimating Equations.
GENLIN Incident_frequency BY Gender Disappear_early Appear_late Persistent
(ORDER=ASCENDING) WITH Problem Age Previous_XP Geekism KBFR_percentage
S_percentage R_percentage K_percentage
 /MODEL Gender Disappear_early Appear_late Persistent Problem Age
Previous_XP Geekism KBFR_percentage S_percentage R_percentage K_percentage
Age*Previous_XP Geekism*KBFR_percentage Geekism*K_percentage INTERCEPT=YES
 DISTRIBUTION=POISSON LINK=LOG
 /CRITERIA METHOD=FISHER(1) SCALE=1 MAXITERATIONS=100 MAXSTEPHALVING=5
PCONVERGE=1E-006(ABSOLUTE) SINGULAR=1E-012 ANALYSISTYPE=3(WALD) CILEVEL=95
LIKELIHOOD=FULL
 /REPEATED SUBJECT=User SORT=YES CORRTYPE=AR(1) ADJUSTCORR=YES COVB=ROBUST
MAXITERATIONS=100 PCONVERGE=1e-006(ABSOLUTE) UPDATECORR=1
 /MISSING CLASSMISSING=EXCLUDE
 /PRINT CPS DESCRIPTIVES MODELINFO FIT SUMMARY SOLUTION WORKINGCORR
 /SAVE RESID.
```

Assumptions

```
* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Pattern Residual_IF
MISSING=LISTWISE REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Pattern=col(source(s), name("Pattern"), unit.category())
 DATA: Residual_IF=col(source(s), name("Residual_IF"))
 DATA: id=col(source(s), name("$CASENUM"), unit.category())
 GUIDE: axis(dim(1), label("Pattern"))
 GUIDE: axis(dim(2), label("Raw Residual of incidence frequency"))
 SCALE: linear(dim(2), include(0))
 ELEMENT: schema(position(bin.quantile.letter(Pattern*Residual_IF)),
label(id))
END GPL.


* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Residual_IF MISSING=LISTWISE
REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Residual_IF=col(source(s), name("Residual_IF"))
 GUIDE: axis(dim(1), label("Raw Residual of incidence frequency"))
 GUIDE: axis(dim(2), label("Frequency"))
 ELEMENT: interval(position(summary.count(bin.rect(Residual_IF))),
shape.interior(shape.square))
 ELEMENT: line(position(density.normal(Residual_IF)), color("Normal"))
END GPL.
```

Disappear early

Generalized estimated equations

```
      * Generalized Estimating Equations.
GENLIN Disappear_early (REFERENCE=FIRST) BY Gender (ORDER=ASCENDING) WITH
Problem Age Previous_XP Geekism Incident_frequency KBFR_percentage
S_percentage R_percentage K_percentage
 /MODEL Gender Problem Age Previous_XP Geekism Incident_frequency
Age*Previous_XP KBFR_percentage S_percentage R_percentage K_percentage
Geekism*KBFR_percentage Geekism*K_percentage INTERCEPT=YES
 DISTRIBUTION=BINOMIAL LINK=LOGIT
 /CRITERIA METHOD=FISHER(1) SCALE=1 MAXITERATIONS=100 MAXSTEPHALVING=5
PCONVERGE=1E-006(ABSOLUTE) SINGULAR=1E-012 ANALYSISTYPE=3(WALD) CILEVEL=95
LIKELIHOOD=FULL
 /REPEATED SUBJECT=User SORT=YES CORRTYPE=AR(1) ADJUSTCORR=YES COVB=ROBUST
MAXITERATIONS=100 PCONVERGE=1e-006(ABSOLUTE) UPDATECORR=1
 /MISSING CLASSMISSING=EXCLUDE
 /PRINT CPS DESCRIPTIVES MODELINFO FIT SUMMARY SOLUTION WORKINGCORR
 /SAVE RESID.
```

Assumptions

```
* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Pattern Residual_DE
MISSING=LISTWISE REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Pattern=col(source(s), name("Pattern"), unit.category())
 DATA: Residual_DE=col(source(s), name("Residual_DE"))
 DATA: id=col(source(s), name("$CASENUM"), unit.category())
 GUIDE: axis(dim(1), label("Pattern"))
 GUIDE: axis(dim(2), label("Raw Residual of problems that disappear
early"))
 SCALE: linear(dim(2), include(0))
 ELEMENT: schema(position(bin.quantile.letter(Pattern*Residual_DE)),
label(id))
END GPL.

* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Residual_DE MISSING=LISTWISE
REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Residual_DE=col(source(s), name("Residual_DE"))
 GUIDE: axis(dim(1), label("Raw Residual of problems that disappear
early"))
 GUIDE: axis(dim(2), label("Frequency"))
 ELEMENT: interval(position(summary.count(bin.rect(Residual_DE))),
shape.interior(shape.square))
 ELEMENT: line(position(density.normal(Residual_DE)), color("Normal"))
END GPL.
```

Appear late

Generalized estimated equations

```
* Generalized Estimating Equations.
GENLIN Appear_late (REFERENCE=FIRST) BY Gender (ORDER=ASCENDING) WITH
Problem Age Previous_XP Geekism Incident_frequency KBFR_percentage
S_percentage R_percentage K_percentage
 /MODEL Gender Problem Age Previous_XP Geekism Incident_frequency
Age*Previous_XP KBFR_percentage S_percentage R_percentage K_percentage
Geekism*KBFR_percentage Geekism*K_percentage INTERCEPT=YES
 DISTRIBUTION=BINOMIAL LINK=LOGIT
 /CRITERIA METHOD=FISHER(1) SCALE=1 MAXITERATIONS=100 MAXSTEPHALVING=5
PCONVERGE=1E-006(ABSOLUTE) SINGULAR=1E-012 ANALYSISTYPE=3(WALD) CILEVEL=95
LIKELIHOOD=FULL
 /REPEATED SUBJECT=User SORT=YES CORRTYPE=AR(1) ADJUSTCORR=YES COVB=ROBUST
MAXITERATIONS=100 PCONVERGE=1e-006(ABSOLUTE) UPDATECORR=1
 /MISSING CLASSMISSING=EXCLUDE
 /PRINT CPS DESCRIPTIVES MODELINFO FIT SUMMARY SOLUTION WORKINGCORR
 /SAVE RESID.
```

Assumptions

```
* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Pattern Residual_AL
MISSING=LISTWISE REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Pattern=col(source(s), name("Pattern"), unit.category())
 DATA: Residual_AL=col(source(s), name("Residual_AL"))
 DATA: id=col(source(s), name("$CASENUM"), unit.category())
 GUIDE: axis(dim(1), label("Pattern"))
 GUIDE: axis(dim(2), label("Raw Residual of problems that appear late"))
 SCALE: linear(dim(2), include(0))
 ELEMENT: schema(position(bin.quantile.letter(Pattern*Residual_AL)),
label(id))
END GPL.


* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Residual_AL MISSING=LISTWISE
REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Residual_AL=col(source(s), name("Residual_AL"))
 GUIDE: axis(dim(1), label("Raw Residual of problems that appear late"))
 GUIDE: axis(dim(2), label("Frequency"))
 ELEMENT: interval(position(summary.count(bin.rect(Residual_AL))),
shape.interior(shape.square))
 ELEMENT: line(position(density.normal(Residual_AL)), color("Normal"))
END GPL.
```

Persistent

Generalized estimated equations

```
* Generalized Estimating Equations.
GENLIN Persistent (REFERENCE=FIRST) BY Gender (ORDER=ASCENDING) WITH
Problem Age Previous_XP Geekism Incident_frequency KBFR_percentage
S_percentage R_percentage K_percentage
 /MODEL Gender Problem Age Previous_XP Geekism Incident_frequency
Age*Previous_XP KBFR_percentage S_percentage R_percentage K_percentage
Geekism*KBFR_percentage Geekism*K_percentage INTERCEPT=YES
 DISTRIBUTION=BINOMIAL LINK=LOGIT
 /CRITERIA METHOD=FISHER(1) SCALE=1 MAXITERATIONS=100 MAXSTEPHALVING=5
PCONVERGE=1E-006(ABSOLUTE) SINGULAR=1E-012 ANALYSISTYPE=3(WALD) CILEVEL=95
LIKELIHOOD=FULL
 /REPEATED SUBJECT=User SORT=YES CORRTYPE=AR(1) ADJUSTCORR=YES COVB=ROBUST
MAXITERATIONS=100 PCONVERGE=1e-006(ABSOLUTE) UPDATECORR=1
 /MISSING CLASSMISSING=EXCLUDE
 /PRINT CPS DESCRIPTIVES MODELINFO FIT SUMMARY SOLUTION WORKINGCORR
 /SAVE RESID.
```

Assumptions

```
* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Pattern Residual_PER
MISSING=LISTWISE REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Pattern=col(source(s), name("Pattern"), unit.category())
 DATA: Residual_PER=col(source(s), name("Residual_PER"))
 DATA: id=col(source(s), name("$CASENUM"), unit.category())
 GUIDE: axis(dim(1), label("Pattern"))
 GUIDE: axis(dim(2), label("Raw Residual of problems that are persistent"))
 SCALE: linear(dim(2), include(0))
 ELEMENT: schema(position(bin.quantile.letter(Pattern*Residual_PER)),
label(id))
END GPL.


* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Residual_PER MISSING=LISTWISE
REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Residual_PER=col(source(s), name("Residual_PER"))
 GUIDE: axis(dim(1), label("Raw Residual of problems that are persistent"))
 GUIDE: axis(dim(2), label("Frequency"))
 ELEMENT: interval(position(summary.count(bin.rect(Residual_PER))),
shape.interior(shape.square))
 ELEMENT: line(position(density.normal(Residual_PER)), color("Normal"))
END GPL.
```

Time on task

Generalized estimated equations

```
* Generalized Estimating Equations.
GENLIN ToT_sec BY Trial Task Task_Completion Gender (ORDER=DESCENDING) WITH
Age ASQ_GEM Geekism Previous_Exp
 /MODEL Trial Task Task_Completion Gender Age ASQ_GEM Geekism Previous_Exp
ASQ_GEM*Geekism Task_Completion*Previous_Exp Age*Previous_Exp Task*ASQ_GEM
Trial*ASQ_GEM Task_Completion*ASQ_GEM INTERCEPT=YES
 DISTRIBUTION=GAMMA LINK=LOG
 /CRITERIA METHOD=FISHER(1) SCALE=MLE MAXITERATIONS=100 MAXSTEPHALVING=5
PCONVERGE=1E-006(ABSOLUTE) SINGULAR=1E-012 ANALYSISTYPE=3(WALD) CILEVEL=95
LIKELIHOOD=FULL
 /REPEATED SUBJECT=Subject SORT=YES CORRTYPE=AR(1) ADJUSTCORR=YES
COVB=ROBUST MAXITERATIONS=100 PCONVERGE=1e-006(ABSOLUTE) UPDATECORR=1
 /MISSING CLASSMISSING=EXCLUDE
 /PRINT CPS DESCRIPTIVES MODELINFO FIT SUMMARY SOLUTION WORKINGCORR
 /SAVE RESID.
```

Assumptions

```
* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Trial Residual_TOT
MISSING=LISTWISE REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Trial=col(source(s), name("Trial"), unit.category())
 DATA: Residual_TOT=col(source(s), name("Residual_TOT"))
 DATA: id=col(source(s), name("$CASENUM"), unit.category())
 GUIDE: axis(dim(1), label("Trial"))
 GUIDE: axis(dim(2), label("Raw Residual of Time on Task (s)"))
 SCALE: linear(dim(2), include(0))
 ELEMENT: schema(position(bin.quantile.letter(Trial*Residual_TOT)),
label(id))
END GPL.
```

```
* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Residual_TOT MISSING=LISTWISE
REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Residual_TOT=col(source(s), name("Residual_TOT"))
 GUIDE: axis(dim(1), label("Raw Residual of Time on task (s)"))
 GUIDE: axis(dim(2), label("Frequency"))
 ELEMENT: interval(position(summary.count(bin.rect(Residual_TOT))),
shape.interior(shape.square))
 ELEMENT: line(position(density.normal(Residual_TOT)), color("Normal"))
END GPL.
```

ASQ scores

Generalized estimated equations

```
* Generalized Estimating Equations.
GENLIN ASQ_GEM BY Trial Task Task_Completion Gender (ORDER=DESCENDING) WITH
Age ToT_sec Geekism Previous_Exp
 /MODEL Trial Task Task_Completion Gender Age ToT_sec Geekism Previous_Exp
Task_Completion*Previous_Exp ToT_sec*Previous_Exp Age*Previous_Exp
Trial*ToT_sec Task*ToT_sec INTERCEPT=YES
 DISTRIBUTION=GAMMA LINK=LOG
 /CRITERIA METHOD=FISHER(1) SCALE=MLE MAXITERATIONS=100 MAXSTEPHALVING=5
PCONVERGE=1E-006(ABSOLUTE) SINGULAR=1E-012 ANALYSISTYPE=3(WALD) CILEVEL=95
LIKELIHOOD=FULL
 /REPEATED SUBJECT=Subject SORT=YES CORRTYPE=AR(1) ADJUSTCORR=YES
COVB=ROBUST MAXITERATIONS=100 PCONVERGE=1e-006(ABSOLUTE) UPDATECORR=1
 /MISSING CLASSMISSING=EXCLUDE
 /PRINT CPS DESCRIPTIVES MODELINFO FIT SUMMARY SOLUTION WORKINGCORR
 /SAVE RESID.
```

Assumptions

```
* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Trial Residual_ASQ
MISSING=LISTWISE REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Trial=col(source(s), name("Trial"), unit.category())
 DATA: Residual_ASQ=col(source(s), name("Residual_ASQ"))
 DATA: id=col(source(s), name("$CASENUM"), unit.category())
 GUIDE: axis(dim(1), label("Trial"))
 GUIDE: axis(dim(2), label("Raw Residual of average ASQ scores"))
 SCALE: linear(dim(2), include(0))
 ELEMENT: schema(position(bin.quantile.letter(Trial*Residual_ASQ)),
label(id))
END GPL.

* Chart Builder.
GGRAPH
 /GRAPHDATASET NAME="graphdataset" VARIABLES=Residual_ASQ MISSING=LISTWISE
REPORTMISSING=NO
 /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: Residual_ASQ=col(source(s), name("Residual_ASQ"))
 GUIDE: axis(dim(1), label("Raw Residual of average ASQ scores"))
 GUIDE: axis(dim(2), label("Frequency"))
 ELEMENT: interval(position(summary.count(bin.rect(Residual_ASQ))),
shape.interior(shape.square))
 ELEMENT: line(position(density.normal(Residual_ASQ)), color("Normal"))
END GPL.
```