Master thesis

Integrating Usability Evaluation in an Agile Development Process

by

Malin Neveryd

LITH-IDA-EX-A–13/068

2014-01-08
Department of Computer and Information Science

Master thesis

Integrating Usability Evaluation in an Agile Development Process

by

Malin Neveryd

LITH-IDA-EX-A–13/068

2014-01-08

Supervisor: Annika Silvervarg, LiU IDA
Erik Claesson, Medius

Examiner: Arne Jönsson, LiU IDA
"It is far better to adapt the technology to the user than to force the user to adapt to the technology." - Larry Marine
Abstract

Medius is a software company which provides IT-solutions that streamlines and automates business processes. The purpose with this thesis was to investigate the possibility to integrate usability evaluation in the development process of Medius software MediusFlow. How such integration would be done, and which usability evaluation methods could be used.

To be able to provide a suggestion, a prestudy was conducted, this in order to get a good overview of Medius as a company as well as the development process of MediusFlow. With the prestudy as a basis, the main study was conducted, and four different usability evaluation methods were chosen. In this study, the four chosen methods were Cognitive Walkthrough, Coaching Method, Consistency Inspection and Question-Asking protocol. These usability evaluation methods were carried out, evaluated and analyzed.

Based on the studies and the literature, a suggestion regarding integration of usability evaluations was made.

The result from this case study was presented as a process chart, where the different phases in Medius software development process are matched together with suiting usability evaluation methods. The relevant phases and their suggested methods:

**Preparation phase** - Cognitive Walkthrough and Coaching Method in combination with Thinking-Aloud and Interviews

**Hardening phase** - Coaching Method in combination with Thinking-Aloud and Interviews, as well as Consistency Inspection

**Maintenance** - Field observation

This result is a part of the overall work towards a more user-centered design of the software.
Preface

This thesis was performed as the final part of my Master’s degree studies in Information Technology at The Institute of Technology, Linköping University. The thesis was conducted in collaboration with the company Medius AB, Linköping and corresponds to 30 ECTS.

While working on this thesis I have had several persons helping me, and I really want to thank all of them. My supervisors Erik Claesson and Annika Silvervarg for all the help and support I have got during this time.

A huge thanks to all the employees at Medius that have helped me with everything, from interviews, to serve as test persons and answering all of my questions.

Last but not least I want to thank my family and friends that have supported me during my education; I could not have done it without you.

Thank you!

Linköping, December 2013
Malin Neveryd
## Contents

### I  Introduction

1  Purpose and aim  
   1.1  Background  
   1.2  Purpose and Research questions  
   1.3  Scope  
   1.4  Disposition  
   1.5  Notation

### II  Theoretical Background

2  Development models  
   2.1  Waterfall Model  
   2.2  Agile development  
      2.2.1  Agile manifesto  
      2.2.2  Scrum  
   2.3  Lean  
      2.3.1  Kanban  
      2.3.2  Scrum-ban

3  User-Centered Design  
   3.1  User Experience  
   3.2  Usability

4  Usability Evaluation  
   4.1  Formative and summative evaluation  
   4.2  Usability Evaluation Process  
   4.3  Usability Metrics  
      4.3.1  ISO 9241  
      4.3.2  Five categories for usability metrics  
   4.4  Usability Evaluation Methods  
      4.4.1  Inspection Methods  
      4.4.2  Testing Methods  
      4.4.3  Inquiry Methods
III  Studies  30

5  Methods  31
   5.1 SWOT-analysis  31
   5.2 Test environments  32

6  Development at Medius  36
   6.1 Method  36
   6.2 Result  37
      6.2.1 Teams and roles at Medius  37
      6.2.2 Medius development process  38
      6.2.3 Interaction Evaluation, what and where  40

7  Usability evaluation in Medius process  41
   7.1 Testing the usability evaluation methods  42
   7.2 Cognitive Walkthrough  43
   7.3 Coaching Method with Interview  44
   7.4 Consistency Inspection  45
   7.5 Question-Asking Protocol  45

IV  Results  47

8  Evaluation Methods  49
   8.1 Cognitive Walkthrough  49
      8.1.1 Execution and post execution interview  49
      8.1.2 SWOT-analysis  50
   8.2 Coaching Method and Interview  50
      8.2.1 Execution  50
      8.2.2 Interviews  51
      8.2.3 SWOT-analysis  52
   8.3 Consistency Inspection  52
      8.3.1 Execution and post execution interview  52
      8.3.2 SWOT-analysis  53
   8.4 Question-Asking Protocol  53
      8.4.1 Execution  53
      8.4.2 Interviews  54
      8.4.3 SWOT-analysis  55
   8.5 General results  55

9  Evaluation suggestion  56
   9.1 Evaluate MediusFlow’s usability  56
   9.2 Evaluation Methods suited for Medius  57
      9.2.1 Preparation phase  57
      9.2.2 Hardening phase  57
      9.2.3 Maintenance phase  58
CONTENTS

9.3 Test persons ............................................. 59

V Completion ....................................... 61

10 Discussion ........................................... 63
  10.1 Usability Evaluations .............................. 63
  10.2 Performing the test evaluations ...................... 63
    10.2.1 Testing protocols ................................ 64
    10.2.2 Selection of metrics .............................. 64
    10.2.3 Scenario design .................................. 64
    10.2.4 Test persons ....................................... 65
  10.3 Reliability and Validity ............................. 65
    10.3.1 Prestudy interviews .............................. 65
    10.3.2 Performance of the test evaluations .............. 65
  10.4 Suggestion ........................................ 66
  10.5 Future work ....................................... 66

11 Conclusion ........................................ 67

A Personas ........................................ 71

B Mobile Scenarios ................................... 73

C Expence Invoice Scenarios ......................... 75

D Test protocol Coaching Method .................... 76

E Test protocol Question Asking Protocol ............ 77
## List of Figures

2.1 The classical waterfall model ........................................... 10
2.2 The Manifesto of the Agile Alliance ................................. 12
2.3 The general structure of the Scrum process (Softhouse, 2006) . 14
2.4 An example of a Kanban board (Crisp, 2013) ...................... 16

4.1 Automated testing versus non-automated testing (Byers, 2013) 23

5.1 SWOT-analysis matrix .................................................. 32
5.2 An example of wireframes used during the first part of the evaluation of the usability evaluation methods .................. 34
5.3 A screenshot of MediusFlow when used during the second part of the evaluation of the usability evaluation methods .... 35

6.1 A sketch of the software development process at Medius AB . 37
6.2 The structure of the development of MediusFlow ................. 38
6.3 A printscreen from Medius virtual kanbanboard, Jira ........... 39
6.4 An overall development map for MediusFlow XI ................. 40

9.1 Suggestion for usability evaluation in Medius’ current process 60
Part I

Introduction
Chapter 1

Purpose and aim

Over the years our society has become more and more dependent on software technology. New software and applications are developed in order to assist us in everyday life and the software companies are competing to create the best program in their category. Because of the strong competition, it is not enough to have the best features; the system needs to have high usability in order to win. The companies have to acknowledge that user-centered design is a big contributor to the success-factor and start to integrate this aspect into their development process. (Nodder & Nielsen, nd)

Out of this need for new and improved technology as well as the shortcomings of the existing development methods, a new type of development method has emerged; the agile software development methods. The agile methods believe that changes are inevitable and that development should embrace changes. The development processes are iterative and relies on simplicity and feedback, in order to guarantee that the desired changes are correct and quickly reflected in the programs. (Jalote, 2008)

1.1 Background

Medius is a Swedish software company, founded in 2001, which provides IT-solutions that streamlines and automates business processes. They develop and sell their own product MediusFlow, a workflow platform, which serves as a support system to existing business systems and compensates for the deficiencies in the business system. (Medius, 2013)

Hollingsworth (1995) defines workflow as ”The computerized facilitation or automation of a business process, in whole or part”. A workflow system is defined as ”A system that completely defines, manages and executes ’workflows’ through the execution of software whose order of execution is
driven by a computer representation of the workflow logic” (Hollingsworth, 1995)).

1.2 Purpose and Research questions

The purpose with this thesis is to examine the possibility of integrating usability evaluation methods and quality requirements in an agile software development process. The specific development process for this investigation is the development process at Medius AB in Linköping, Sweden.

In order to understand how this integration can be done, a case study will be performed at Medius AB. According to Runesson & Höst (2008) a case study is a suitable research methodology for research in software engineering. This is because a case study studies the contemporary phenomena in its natural context.

To be able to answer the question regarding the integration of usability evaluation, the current development process needs to be examined. The goal of the case study is to provide Medius with various usability evaluation methods that they can use in their development process in order to get feedback on the usability of their system MediusFlow. In order to conduct a case study with the purpose and aims described in the section above, the following question needs to be answered:

- How can usability evaluation methods be integrated into an agile development process?
- Which methods can be used and how are they used?

1.3 Scope

This case study will only involve non-automatic usability evaluation methods, more about this constraint can be read in section 4.4.

This study has a time limit of 20 weeks. Thus only a selection of usability evaluation methods has been investigated thoroughly and only one usability evaluation iteration could be performed and evaluated.

The study is based on Medius AB and their software development process and their goals towards an integration of usability evaluation in this process. The focus is on the teams and the work around the development of their product MediusFlow, therefore the information about the company will only describe those parts. The results of this case study will not be general, they will be adapted to Medius as a company and their software development process.
1.4 Disposition

This report is divided into six different parts: Introduction, Theoretical Background, Studies, Results and Completion.

Part I, Introduction, is the introduction to this case study and contains background to the study, purpose and scope.

Part II, Theoretical Background, is the theoretical background, containing three chapters regarding different development models, user-centered design and usability evaluation.

Part III, Studies, contains the two studies that were made as well as a description of how they were performed and what they included. The results from the prestudy are also presented in this part.

Part IV, Results, presents the results from the main study considering the different usability evaluation methods and the suggestion made.

Part V, Completion, serves as the closure of the case study, and contains two chapters, the discussion and the conclusion chapters. The discussion discusses the results of the case study and the conclusion chapter presents the conclusions and answers the research questions.

1.5 Notation

The following abbreviations will be used in this thesis:

GA - General Availability
ISO - International Organization of Standardization
OTS - Off-The-Shelf, referrers to software that is pre-built.
R&D - Research and Development
RTM - Release To Market
UCD - User-Centered Design
UI - User Interface
UX - User eXperience
Part II

Theoretical Background
Chapter 2

Development models

This chapter describes the theoretical background that is the foundation of the case study. The chapter will provide the reader with an introduction to the world of different development models.

2.1 Waterfall Model

The waterfall model was proposed by Winston. W. Royce in 1970. The name "Waterfall" came later and originates from the fact that the model is a sequential development method, in which the process is flowing like a waterfall on a downhill. (Jalote, 2008)

The waterfall model includes several different phases, and each phase needs to be completed before moving on to the next phase as seen in figure 2.1. Bassil (2012) presents the five phases in the waterfall model as:

- Analysis
- Design
- Implementation
- Testing
- Maintenance

*Analysis phase:* This phase is also called the software requirement specification (SRS): The customer needs are identified and documented. In this phase both functional requirements such as purpose and software attributes, as well as non-functional requirements like design requirements are defined by the system and business analyst. An example of an outcome of this phase is a requirements document. (Bassil, 2012; Jalote, 2008; Petersen, Wohlin & Baca, 2009)
*Design phase:* The design phase includes the process of defining and planning a software solution. One of the outcomes from this phase is design documents. (Bassil, 2012; Jalote, 2008)

*Implementation phase:* This is the phase where the software code is written and compiled; it is where the requirements is converted into a production environment. The output from this phase is the final code. (Bassil, 2012; Jalote, 2008; Petersen et al., 2009)

*Testing phase:* This phase is also known as verification and validation, which is exactly what it includes. Verifications are made in order to check if the requirements are fulfilled and that it achieves its original purpose. Examples of outcomes from this phase are test reports and test plans. (Bassil, 2012; Jalote, 2008; Petersen et al., 2009)

*Maintenance phase:* It is the phase after the product has been released and in which potential bugs are corrected, performance is improved as well as some improvements on the quality is made. (Bassil, 2012; Petersen et al., 2009)

According to Jalote (2008) the main idea with these phases is to divide the different concerns into the different phases. This is done so that the big
task of building software is broken down into smaller, more manageable
tasks. The idea with *separation of concerns* is one of the things that make
the model simple.

Even though the model is widely used it has some serious drawbacks such as;

- It assumes that the requirements are fully determined before the
design phase, having unchanging requirements are not always
realistic.

- It only delivers what is specified in the requirements. This in
combination with inflexible requirements can make the users and
stakeholders add features that may come in handy, but might never
be needed.

- It uses the ”big-bang” approach, in other words, the whole software
is delivered in one big delivery at the end.

- The model is a document-driven process and requires formal
documentation at the end of each phase.

2.2 Agile development

Agile development is a family of incremental and iterative development
methods. This family of methods is a backlash from the waterfall model,
and tries to solve the problems that are listed in the previous section.

There are several different agile methods, such as Scrum, Extreme
programming (XP), Crystal and so on. All of the agile methods share their
main values; these are described below in the Agile manifesto section.
(Cockburn, 2007)

2.2.1 Agile manifesto

The agile software development manifesto was created early in 2001 by
representatives from different agile development methods calling
themselves ”the agile alliance”. The manifesto consists of four core values
as seen in the figure 2.2 from the agile manifesto web page
(TheAgileAlliance, 2001);
Behind the manifesto there are twelve principles, these are the characteristics that distinguishes agile practices from heavyweight processes: (Martin, 2003; Cockburn, 2007)

- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- Business people and developers must work together daily throughout the project.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
• The most efficient and effective method of conveying information to
and within a development team is face-to-face conversation.

• Deliver working software frequently, from a couple of weeks to a
couple of months, with a preference to the shorter timescale.

• Continuous attention to technical excellence and good design
enhances agility.

• At regular intervals, the team reflects on how to become more
effective then tunes and adjusts its behavior accordingly.

• Simplicity — the art of maximizing the amount of work not done — is
essential.

• Working software is the primary measure of progress.

The Agile Manifesto does not form a development method but provides
important core values and principles. One development method that
follows the Manifesto is the scrum methodology, which is described further
below.

2.2.2 Scrum

Scrum is a framework and was originally designed for new software
development for a competitive market. The name ”scrum” comes from
rugby scrums and the fact that everyone is pushing together in a mutual
mission. (Hughes & Cotterell, 2009) An overviewing figure of the scrum
process is presented in figure 2.3

The development process in scrum consists of sprints; each sprint often
lasts between one and four weeks but the general idea is to keep the same
iteration length over a period of time. The sprints are the development
phases, time boxed iterations. At the beginning of each sprint, a plan is
made, a sprint backlog. This sprint backlog consists of the user stories and
their associated tasks that will be carried out during that sprint. These
stories comes from the product backlog which is a prioritized list of all the
desired deliverables for the product. These deliverables are often known as
items, but can also be called user stories. According to Sims & Johnson
(2012) every items or stories should include:

• A brief description of the desired functionality

• Which users the story will benefit

• The reason why this story is valuable

• An estimation of how much work the story takes to complete
• Acceptance criteria that indicates when the story has been implemented correctly.

After the sprint is done, the stories that are in the sprint backlog should be done and are defined as done if they fulfill the acceptance criteria. The stories that are done is recorded on the burn down chart, which is a graph where time is on the x-axis (preferably the sprints) and scope is on the y-axis (number of stories). This burn down chart gives the team a good overall view over the development progress.

Figure 2.3: The general structure of the Scrum process (Softhouse, 2006)

The user stories are recordings of the requirement and they are written by the product owner. The product owner is the owner of the product backlog and is in charge of the prioritization of the stories in the product backlog. This person also represents both the customers as well as the interests of the business. (Sims & Johnson, 2012)

Another role in Scrum is the scrum master, which is the expert and advisor about Scrum and also works as a facilitator. The scrum master helps the team members to learn and also apply scrum, and is available to help the team remove road-blocks that are stopping them from doing their job. (Sims & Johnson, 2012)
2.3 Lean

The Lean concept was first known as the Toyota Productions System or Just-in-Time and was founded by Taiichi Ohno. Lean manufacturing is production of goods and focuses on preserving value with less work, where value is defined as any process or action a customer is willing to pay for, for example, there should be less waste, less human effort and so forth. It centers on production process but it can also be viewed as a management technology for manufacturing cost-reduction. In lean there are different tools that help to eliminate waste in the different areas of the production; an example of one such tool is Kanban. (Wang, 2011)

2.3.1 Kanban

The Japanese word Kanban translates to ”signal card” and has become a synonym to demand scheduling (Wang, 2011). Kanban is a part of a lean initiative to morph the culture of organizations and encourage continuous improvement (Kniberg & Skarin, 2010).

Kanban is not a software development, project management lifecycle or a project management process. It is an approach to introduce change to an existing software development lifecycle, a signaling system to trigger action. (Kniberg & Skarin, 2010; Wang, 2011)

Kanban uses a visual control mechanism to track work as it flows through the various stages of the value stream. The main points of Kanban are; (Kniberg & Skarin, 2010)

- Visualize the workflow
  - Split the work into pieces, write on a card and put it on the wall
  - Use name columns to illustrate where each item is in the workflow
- Limit WIP (Work In Progress) - assign explicit limits to how many items may be in progress at each workflow state
- Measure the lead time (average time to complete one item, sometimes called ”cycle time”), optimize the process to make lead time as small and predictable as possible.
The Kanban board, as seen in figure 2.4, is a good way of visualizing the workflow and at the same time limit the work in progress.

![Figure 2.4: An example of a Kanban board (Crisp, 2013)](image)

### 2.3.2 Scrum-ban

Scrum-ban is a mixture of both Scrum and Kanban. The approach with sprints that Scrum has does not suit all companies, but maybe some other aspects of Scrum may work great for the company. To take what is wanted and needed from Scrum and mix it into Kanban is not uncommon (Kniberg & Skarin, 2010).
Chapter 3

User-Centered Design

In order to understand the term usability, a good understanding of user-centered design and user-experience is needed. These terms are described below and are followed by a small explanation of the term usability.

Bias and Mayhew (2005) defines user-centered design (UCD) as a comprehensive development methodology which is driven by clearly specified, task-oriented business objects and the recognition of user needs; limitations and preferences. When and if UCD is correctly used, this approach meets both the users as well as the business needs.

This definition is according to Gullriksen and Göransson (2002) based on twelve core principles:

- **User focus** – the businesses goals, users’ job assignments and needs should guide the development early.

- **Active user involvement in the development** – representative users should participate, early and continuous during the systems life cycle.

- **Evolutionary development** – the system should be developed iteratively and incrementally.

- **Common and shared understanding** – the design should be documented with a comprehensible representation for everyone who is involved.

- **Prototyping** – the prototypes should be used early and continuously in order to visualize and evaluate ideas and design solutions with the end users.

- **Evaluate real use** – measurable usability goals and criteria for the design should as far as possible guide the development.
• **Explicit and declared design activities** – the development process should contain dedicated and reflective design activities.

• **Multidisciplinary teams** – the development should be performed by efficient teams with a spread of skills.

• **Usability advocate** – experienced usability advocates should be involved early and continuously during the entire development project.

• **Integrated system design** – every part that affects the usability should be integrated with each other.

• **Locally adapt the processes** – the user-centered process should be specified, adjusted and introduced locally in every organization.

• **User-centered attitude** – a user-centered attitude should always be established.

Some of these principles build on the terms user experience and usability. These terms are further described in following sections.

### 3.1 User Experience

The term user experience and consumer experience is sometimes used interchangeably, but this is not accurate.

The International Organization of Standardization (ISO 9241-210) defines user experience as "a person’s perceptions and responses that result from the use or anticipated use of a product, system or service."

When talking about user experience, the experience is limited to the actual usage of your product and not the experience the user get when for example, looking it up on the internet. Consumer experience covers the product usage as well as getting it and perhaps replacing it with another product. (Kraft, 2012)

### 3.2 Usability

The term usability is a widely used term and as any other term that is frequently used, there are different definitions of the term.

Usability is defined by The International Organization of Standardization (ISO 9241-11) as: "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." (Bias & Mayhew, 2005)

Nielsen (1993; 2012) defines usability as a quality attribute that assesses how easy user interfaces are to use and that usability has multiple
components. That it is associated with five usability attributes; Learnability, Efficiency, Memorability, Errors and Satisfaction.
Chapter 4

Usability Evaluation

Usability Evaluation is comprised of different methods for measuring the usability of a system and for identifying specific problems with the interface. The usability evaluation is an important part of the overall usability process and is a process of its own. (Ivory, 2001)

Evaluation should occur throughout the design life cycle, with the results of the evaluation feeding back into modifications to the design. This process involves several activities that are method dependent. (Nielsen, 1993)

4.1 Formative and summative evaluation

Faulkner (2000) cites Hewitt (1986) and his two types of evaluation, formative and summative.

A formative evaluation is made in order help the design process and is used to improve and formulate the design. It involves a close work with the users of the system and collecting feedback about the system from these users.

A summative evaluation is on the other hand much probable to require quantitative data. According to Faulkner (2000) this evaluation is of the overall performance of the user and the system, as the usability and the effectiveness of the system.

When comparing the evaluation methods, in order to choose only one to use, Faulkner (2000) says: ”It is not possible to ignore one form of evaluation in favor of another. It would be like driving a car with only forward or reverse gear - but not both!”
4.2 Usability Evaluation Process

The process of evaluating usability can involve different activities for example: (Ivory, 2001; Faulkner, 2000)

1. **Specifying the usability evaluation goals** - depending on where in the usability interface life cycle the usability evaluation is applied, different usability evaluation goals are relevant.

2. **Determine which UI aspects that should be evaluated** - if the usability interface is big and complex, it may not be possible to evaluate the whole interface. If specific aspects of the interface are chosen, these aspects must be consistent with the goals of the evaluation.

3. **Identifying the target users** - it is important to decide which user characteristics that are the most relevant for the evaluation and the chosen user interface aspects.

4. **Select which usability metrics that should be used** - these metrics are essential for the evaluation. ISO 9241 recommends using; Effectiveness, Efficiency and Satisfaction. Further reading about usability metrics is found in section 4.3.

5. **Select which evaluation method/methods that should be used** - choose from five different classes: inspection, inquiry, analytical modeling, usability testing and simulation. The different methods differ in many dimensions, such as: costs, applicability and results.

6. **Select tasks that should be evaluated** - the tasks chosen must reflect the usability interface aspects of the system, the users and the chosen usability evaluation method.

7. **Design the experiments** - the evaluator needs to decide on the number of participants, users and evaluators, the evaluation procedure and on the system and environment setup.

8. **Capture usability data from the experiment** - the evaluator uses the chosen usability evaluation method to record the usability metrics. Depending on the method, the evaluator might also record usability problems.

9. **Analyze and interpret the captured data** - summarize the results and parse it.

10. **Critique the UI and suggest improvements** - illustrate the user interface flaws and how to improve the design.

11. **Present the results from the evaluation** - inform the results from the evaluation process to the stakeholders.

The analysis and interpretation may indicate that the usability evaluation needs to be repeated, if so, the process starts over from the top.
4.3 Usability Metrics

When measuring or evaluating, metrics are used and the usability field is no different. Metrics are based on a reliable system of measurement: using the same measurements every time something is measured. In the case of usability metrics, they are measuring some aspect of user experience. These usability metrics must be observable in some way, directly or indirectly, and also quantifiable. (Tullis & Albert, 2008)

4.3.1 ISO 9241

As mentioned in previous section, ISO 9241 recommends using effectiveness, efficiency and satisfaction. They define these terms accordingly:

Effectiveness; is the accuracy and completeness of users goals. Examples of effectiveness metrics are: functions learned, workload and percentage of goals achieved.

Efficiency; assesses resources extended in relation to accuracy and completeness of user goals. Examples of efficiency metrics are: learning time, percent or number of errors and time to complete a task.

Satisfaction; reflects the absence of discomfort for the user and positive attitudes about use of an interface. Examples of satisfaction metrics: ease of learning and error handling. (Ivory, 2001)

4.3.2 Five categories for usability metrics

Tullis & Alfred (2008) divides usability metrics into five different categories and describes them as following:

Performance Metrics, these metrics are calculated based on specific user behaviors as well as the use of tasks or scenarios. Performance metrics are the best way to evaluate the efficiency and the effectiveness of many different products as well as knowing how well the users are actually using a product.

Issues-Based Metrics, are good when identifying user interface issues. The most useful issues are those that point out possible improvements in the system. Three things that can help to identify usability issues are; Understanding what is (and is not) an issue, being very observant of the test persons behavior and knowing both the system and the usability problems that arise.

Self-Reported Metrics, is when the users and participants are asked for information. These metrics are sometimes called subjective data and
preference data. Self-reported metrics are the most obvious way of learning about the usability according to Tullis & Alfred (2008).

Behavioral and Physiological Metrics, are measuring the test persons behaviors, conscious and unconscious behaviors, verbal and non-verbal. Body language in form of facial expressions and finger drumming on the table are some metrics that can be observable.

Combined and Comparative Metrics, combined metrics is when two or more metrics are combined and analyzed together so that new data can be derived. Comparative metrics are on the other hand when two or more metrics are compared and analyzed in order to extract new information from existing data.

4.4 Usability Evaluation Methods

Usability evaluation models are classified into five different categories: Inspection, Analytical Modeling, Simulation, Inquiry and Testing, and are either automated or non-automated. As seen in figure 4.1 automated tests are not economic when only running it a few times. Since Medius is in the starting blocks of using usability evaluations to evaluate their software, a restriction has been done to only discuss and evaluate non-automatic evaluation methods. Another restriction made is, that in order to get started with usability evaluations it is easier to use the same methods to both the computer software and their mobile applications. Therefore no analytical evaluation models will be discussed. Neither will simulation methods be discussed, there are no non-automatic models in this classification and thereby not further discussed.

Figure 4.1: Automated testing versus non-automated testing (Byers, 2013)
4.4.1 Inspection Methods

According to Mack & Nielsen (1994) usability inspection is a generic name for a number of evaluation methods where the evaluation of the user interface is based on the considered judgment of the inspector(s).

Typically, an inspection is used when a user interface had been generated, and its usability needs to be evaluated from the user’s point of view.

Cognitive Walkthrough

The cognitive walkthrough is a method where the expert evaluators construct scenarios from either the specification or an early prototype. One of the evaluators plays the role of a user and attempts to simulate the user’s problems when “walking through” the interface, acting like the interface was built and was trying to perform a task. Usually the main focus with this approach is to see how easy a system is to learn, it is a kind of exploratory learning. (Mack & Nielsen, 1994)

When performing a cognitive walkthrough it is good to start with evaluating the system specification, from a user task point of view, in order to identify the purpose and the goals of the tasks. During the walkthrough it is important to identify problems with reaching the goals. (Dix, Finlay, Abowd & Beale, 2004)

This method is good to use early in the development process because of the fact that it can be applied on only the system specifications.

Pluralistic Walkthrough

The pluralistic walkthrough is a variation of the cognitive walkthrough and a traditional walkthrough, where the users, the evaluators and the developers inspect the interface as a group. The goal with this method is to walk through the interface and discuss potential interface flaws.

When carrying out a pluralistic walkthrough all the participants are assuming the role as the user and trying to performing the tasks by carrying out each of the step in the process. Before discussing decisions, the evaluators are writing down their chosen action and only after this is done, they may discuss their answers.

In order to get the most out of this technique it should be performed at an early stage of the development process. This because of the feedback from this method is often user preferences and opinions. (Hom, 1998)
Heuristic Evaluation

In this inspection method the evaluators independently evaluate and inspect the interface. In general, heuristic evaluation is done by many evaluators, because of the fact that one evaluator will never be able to find all usability defects by himself. After all evaluations have been finished, the evaluators are allowed to talk to each other and compile their results. (Mack & Nielsen, 1994; Hom, 1998; Dix et al., 2004)

According to the creators of the heuristic evaluation Nielsen and Molich, it is important to have several evaluators, a group of three to five evaluators is sufficient. With five usually resulting in about 75% of the overall usability problems being discovered.

This model is suited to use early in the usability engineering life cycle, but can be used almost any time in the development cycle. The experts can use paper mockups or maybe just the design specification and find a lot of usability problems before they are implemented. (Mack & Nielsen, 1994; Hom, 1998; Dix et al., 2004)

Consistency Inspection

During a consistency inspection the expert(s) verify the consistency across a family of interfaces. The terminology, fonts, color schemes, layout, input and output formats, will be checked, both regarding the interfaces as well as the documentation. With the usability analysis as a basis, an evaluation team meets and negotiates and decides on the implementation.

To carry out this evaluation method, an inspection team will be necessary. The team should consist of members from each development team for all parts covered by the inspection. For this to work, the team members need to have the authority to negotiate and perhaps change the design of the part they are representing.

The consistency inspection is most suitable to use when the development process has not come to the point that the software code may need to be comprehensively substituted. (Hom, 1998; Schneiderman & Plaisant, 2010)

Feature Inspection

This inspection method focuses on the functionality provided by the software. It does not necessarily just involve the functionality but can also involve the design of that particular functionality. (Mack & Nielsen, 1994)

In order to use feature inspection one needs to list the system features in the order they would be used to be able to perform various tasks. These tasks are both typical assignments, steps that would not be natural for the users as well as steps that require extensive knowledge about the system (Nielsen, 1993). Then asking the questions; Are the features well named
and easily recognized? Can a user get to each feature without much trouble? (Hom, 1998)

This evaluation technique is best suited in the middle-part or at the end of the development process, when the functions of the software are known.

4.4.2 Testing Methods

User testing with real users is the primal usability method, because of the fact that it provides direct information that describes how people use computers and what their problems are; this is if the concrete interface is being tested. (Nielsen, 1993)

Thinking Aloud Method

During this method/test technique the users are asked to think-aloud and try to express how they feel, their opinions, what they are doing and why they are doing it while completing a task. (Faulkner, 2000)

To evaluate software with this method, the user needs to be provided with the software as well as a scenario/task to perform (Hom, 1998).

The think-aloud method is used in order to get a better understanding of the participant’s mental model during the interaction with the interface. It is a method that can be used at any stage in the development process, and is a very cheap way of getting lots of good quality feedback during usability testing. (Hom, 1998; Mack & Nielsen, 1994)

Coaching Method

During a coaching evaluation the participants are allowed to ask system-related questions to an expert coach. The expert coach is often the evaluator, but in some cases a separate evaluator acts as a coach. These different approaches can provide different results from the evaluation. (Hom, 1998)

The main goal with this evaluation technique is to provide better training and documentations and eventually redesigning the interface. This based on the questions that was asked from the tester, this to eliminate the need for questions. (Hom, 1998; Ivory, 2001)

Co-discovery Learning

This usability evaluation method involves two users at the same time. They will try to solve the tasks together while being observed by the evaluator. To evaluate software with the co-discovery method, the user needs to be provided with the software under test as well as a scenario/task to perform. The users are instructed to help each other as
they would do an ordinary day at work and at the same time explain what they are thinking about. (Ivory, 2001)

The co-discovery method is best suited for Computer-Supported Cooperative Work (CSCW) software and can be used anywhere in the software development cycle. (Ivory, 2001)

**Performance Measurement**

In this method focuses on collecting quantitative data of the users’ performance when completing a task. This method can be combined with another method in order to collect qualitative data as well. When choosing performance measurements it is important to consider that the objectives must be quantifiable, the experimental design is really important and data does not tell the whole story. (Nielsen, 1993; Hom, 1998)

**Question-asking Protocol**

This method is an extension of the Thinking-Aloud protocol discussed above. The protocol allows the testing team to ask the participants questions about the interface. The aim with this technique is to get a better understanding of the participant’s mental model of the software. (Ivory, 2001)

### 4.4.3 Inquiry Methods

The inquiry method family consists of evaluation methods that require feedback from the users. These methods are often used in a usability test, as a complement to the test. The goal of these methods is to gather subjective impressions of the user interfaces, and sometimes to gather auxiliary data after a system release.

**Interviews**

An interview about the users’ perception of the software offers a direct and structured way of assembling information. One of the main benefits with interviews is that the level of the questions can be adjusted to that particular purpose and user. To be able to gain as much as possible from the interviews, they need to be thoroughly prepared. (Hom, 1998; Dix et al., 2004)

Interviews can be used to reveal problems that had not been anticipated or have not occurred under an observation. This technique is good to obtain information about user preferences, impressions and also the users’ attitude towards the software. (Dix et al., 2004)

There are different types of interview, unstructured, semi-structured and structured. In the first, the the questions are free and no script is followed.
In the lateral, a script is used and the order of the questions are important. No question that is not in the script are allowed to be asked. In a semi-structured interview, there is a script but the questions do not have to be asked in a specific order and additional questions can be asked. (Jacobsen, 1993)

Field Observation

A field observation is often done with released software. The evaluators visit the customers’ workplace and observe the customers while they are using the software under evaluation. This observation allows the evaluators to understand how the users actually use the software to perform the day-to-day tasks. During the field observation the evaluator can use an unstructured or a structured interview to get to know the user. For example how they use the product for a specific task. (Hom, 1998; Dix et al., 2004)

Focus Groups

A focus group is when six to nine users meets and discusses their perceptions, including issues relating the software. The evaluator acts as a moderator, and collects the different opinions from the discussion. This method is used in order to better understand the users’ thoughts about the software. (Hom, 1998; Ivory, 2001)

Contextual Inquiry

A contextual inquiry is a kind of field observation where the evaluator studies the user in context for two to three hours at the users’ workplace. The main difference from field observations as described above, is that the contextual inquiry is more of an investigation process. (Hom, 1998) According to Hom (1998) this method is based on three core principles: understanding the context in which a product is used, that the usability design process must have a focus and that the user is a partner in the design process. The investigator is not a passive observer, it is the investigators job to gain a shared understanding of how the work happens and interpret and understand what they really mean.

This method is best used when the software has been released and the evaluator can observe and investigate the usage of the software. (Hom, 1998)

User Feedback

This evaluation method is when the software users expresses their opinions whenever necessary or at their own convenience. This method is used on already released software and can be done by having a feedback button in
the user interface. Other ways of collecting user feedback is per email or via a web site. (Ivory, 2001)
Part III

Studies
Chapter 5

Methods

This chapter reasoning and discusses about the approaches and methods used in this case study.

The aim with this thesis was to test and suggest different usability evaluation methods that would suit Medius as a company as well as their development process. Therefore it was important to get a good overview over their development process. This to get an accurate view on how Medius works and which kinds of evaluation methods would suit their needs. To get this overview, a small prestudy was made. The method used and the results from the prestudy is presented in chapter 6.

The main study is performing four different usability evaluation methods and then comparing them to each other and evaluating them with Medius needs as basis. The results from the prestudy were used in the decision making of which usability evaluation methods were going to be tested. More about each study can be read in the sections that follow.

The four evaluation methods were chosen together with Medius user experience expert and these decisions where based on Medius needs, their assets in form of personnel as well as the prestudy.

5.1 SWOT-analysis

A SWOT-analysis is a structured planning method in which Strengths, Weaknesses, Opportunities and Threats are evaluated. A SWOT-analysis can be resembled with a 4x4 matrix as shown in figure 5.1. This analysis can be used in many different areas but it is mostly used in business related subjects such as projects, products and companies. (Fahy & Jobber, 2012)
For each of the four usability evaluation methods that was evaluated, there was a SWOT-analysis made. This was in order to facilitate the comparison of the different methods and be able to make an evaluation suggestion for Medius.

![SWOT-analysis matrix](image)

**Figure 5.1: SWOT-analysis matrix**

### 5.2 Test environments

When evaluating systems in an early stage, it is often done by using for example mockups, prototypes or wireframes. During this case study, both wireframes and the actual system was used to evaluate the user interface. Figure 5.2 shows some examples of wireframes used in the testing, and figure 5.3 shows a screenshot of MediusFlow.

Wireframes is a tool used to simply create and visualize design, and is a sketch of the placements of the different elements. The detail level of the design is varying in different wireframing techniques, for example pen and
paper or a computer program can be used. (Vegh, 2010) In computer programs it is common that one uses "drag-and-drop" functions, the objects that are wanted is dragged from a menu.

In this case study and at Medius, the wireframes are made with Balsamiq Mockups, which is a computer program with "drag-and-drop" functions.
Figure 5.2: An example of wireframes used during the first part of the evaluation of the usability evaluation methods
Figure 5.3: A screenshot of MediusFlow when used during the second part of the evaluation of the usability evaluation methods.
Chapter 6

Development at Medius

To get a good overview and understanding of Medius as a company, as well as their development process, a small prestudy was conducted.

6.1 Method

The overview was generated by the conduction of semi-structured interviews with employees at the Linköping and the Stockholm office. The semi-structured approach was chosen over the structured and the unstructured. This in order to let the interviewees talk freely, but at the same time, make sure that the questions the interview was to answer, was answered. The questions asked were:

- What is your role at Medius?
- How does the overall development process look like today?
- Where in the development process is the software testing performed?
- What do Medius want to get out of the usability evaluations?
- Where in the development process do you think that usability tests are needed?
- Who should the usability test be on?

The interviewees where chosen by their roles, either in the development process or in the overall structure of the company. The interviewees’ role involves work with the overall process, but from different angles, and are: the User Experience Manager, the two Product Owners, a Product Manager and the R&D Director. More about the roles is found in subsection 6.2.1.
6.2 Result

Based on the interviews that were conducted, a process map over the development process was created, see figure 6.1, as well as an overview of the roles that are associated with the software development process. The map provides a rough overview over Medius development process, although a more describing and profound explanation of their development process is given below.

![Diagram of software development process at Medius AB]

Figure 6.1: A sketch of the software development process at Medius AB

6.2.1 Teams and roles at Medius

MediusFlow is divided into a platform and an application part as shown in figure 6.2. The people who work with these parts are either in the Application tribe, the Platform tribe or both. The different roles in the tribes are: Product Owner, Lead Programmer, Developer, Business Systems Analyst, Quality Assurance (QA) and User Experience (UX). These tribes are further divided into squads, which includes a Team Leader, a QA, a Lead Programmer and Developers. Other roles that are connected to MediusFlow but are not in either platform or the application part are: Product Manager and R&D Director.
6.2.2 Medius development process

MediusFlow is a kind of off-the-shelf software with high configurability, built for a generic customer. One can say that Medius, as well as the buyers of MediusFlow, are the customers, because of the facts that the software is generic and Medius uses their own software MediusFlow. When deciding the new version in MediusFlow, Medius enters a sort of preparation phase. Their product council, consisting of representatives from the R&D, Sales, Marketing and Product Management, creates official and unofficial roadmaps. These consist of one short term (6 months) and one long term (3 years), and needs to be approved by the management. The roadmaps contain different initiatives, which can be new applications to MediusFlow but it can also be areas that needs improvements in current applications or the platform.

The product council votes on the priority order of the different initiatives. To be able to choose which of the initiatives that can be implemented in this process, the initiatives are time estimated and a prestudy is made. The initiatives that are not chosen are kept in the product backlog and are reprioritized at the next product council meeting.

The initiatives that were chosen are then broken down into main requirements by the product owner for the application and the product owner for the platform among others. The main requirements are put into a Jira, Medius virtual Kanban board, seen in figure 6.3. The main requirements are divided into smaller and more detailed requirements (user stories) and put into another Jira and then further divided into tasks. The
Figure 6.3: A printscreen from Medius virtual kanbanboard, Jira
limits of each workflow state is marked in the red square and the number in the white box is number of tasks in the state. The states that are red are states that contain too many tasks. The pictures at the right side of the task are the person that is assigned to that task.

The system is released at an early stage to so-called early adopters, which are companies that implements the software before it is fully implemented. Meanwhile the hardening phase starts and the development continues to stabilize the software before the general availability. When MediusFlow is available generally the development process turns into the last phase, the maintenance phase.

6.2.3 Interaction Evaluation, what and where

Two of the questions in the interview was ”What do Medius want to get out of the usability evaluations?” and ” Where in the development process do you think that usability tests are needed? ”.

All the respondents completely agreed on that the goal with the usability evaluation should be to get feedback on the system as early in the process as possible. Results they would want from usability evaluations were such as: uniformity in the system, enhanced learnability of the system, a more intuitive system. These results falls under the usability metric categories; self-reported, issues-based and performance metrics.
Chapter 7

Usability evaluation in Medius process

When testing software, it is done in different steps in the software life-cycle. Acceptance testing verifies the requirements, system testing verifies the system design, integration testing verifies the interactions of the modules and unit testing verifies the implementation. (Copeland, 2004) Testing the user interface is no different; A decision of how extensive the tests are going to be, is needed. Should the interface tests only be of the overall system, or should it be done when integrating the units?

Together with Medius user experience expert, it was decided to evaluate the user interface at an initiative and requirement level. We decided to test the user interface both before it is implemented, and when the system is in a late stage of the development process. These decisions were both based on the user experience expert’s opinions as well as the results from the prestudy, where all of the interviewees wanted to get feedback on the system as early as possible.

When choosing evaluation methods to evaluate and test, all of the methods where compared to each other and studied thoroughly. Some of the methods were eliminated from the selection at an early stage; others were eliminated from the selection in this step. The methods that needed more than one user interface expert were removed from the selection because of the fact that Medius currently has just one user interface designer. Methods like performance measurement and feature inspection were not interesting because of the fact that the results from these methods did not match the desired results. Others were not chosen because of the similarity with other already chosen methods. Another criteria for the evaluation methods to fulfill was ease of use and that the method was suited for evaluating software in the early stages of its life-cycle as well as software in the latter parts of the life-cycle. The chosen
methods were not to be too similar either.

The final methods that were chosen to evaluate and test were; Cognitive Walkthrough, Coaching Method in combination with Interview, Consistency Inspection and Question-Asking Protocol. The first two methods were used early in the development process and the two latter were used in a later stage in the development process.

7.1 Testing the usability evaluation methods

The testing of the different methods was divided into two parts, one part with an initiative that were in mock-up phase, the MediusFlow iPhone application, the other part was on the feature in MediusFlow called expense invoice that is in a continuous improvement phase. The methods that were tested in the first part were Coaching Method with Interview and Cognitive Walkthrough and in the other part were Question-Asking Protocol and Consistency Inspection. When preparing to perform and evaluate the following steps were used as a guideline, more about these steps can be found in section 4.2.

1. Specifying the usability evaluation goals - These goals were based on the results from the prestudy, and were the same for all methods.
   - Improve the usability of the different applications in MediusFlow.
   - Find potential interface issues early.
   - Enhance the ease of learning MediusFlow.

2. Determine which UI aspects that should be evaluated - As written above, the requirements of each initiative or feature was to be evaluated.

3. Identifying the target users - These target users was picked out from the Medius’ personas, Peter and Anna. These personas can be found in appendix A. The first part had both Peter and Anna as target user; the second part had only Anna according to the requirements.

4. Select which usability metrics that should be used - The selection of metrics was based on the results from the prestudy. The metric categories that were interesting were: issues-based, self-reported and performance.

5. Select which evaluation method/methods that should be used - This selection is written more in detail above. The chosen methods was; Cognitive Walkthrough, Coaching Method with Interview, Question-Asking Protocol and Consistency Inspection.
6. **Select tasks that should be evaluated** - These tasks were created according to the requirements of the initiative or feature.

7. **Design the experiments** - For the first part a set of wireframes were created to fit the tasks, examples of these wireframes can be found in figure 5.2. For the second part the actual system was set up according to the tasks, screenshots of this can be found in figure 5.3.

8. **Capture usability data from the experiment** - This was the performance of the evaluation methods.

The rest of the steps in the example in section 4.2 were not relevant according to the scope of this thesis and was not further considered. However, the feedback on the system that was the result from the evaluations was further considered by the user experience expert at Medius.

After the completion of every evaluation of the methods, a semi structured interview was held with the user experience expert at Medius. The questions asked were:

- What is your perception of this method? Easy/difficult to perform?
- Is there any obvious advantages?
- Is there any obvious disadvantages?
- Do you think the fact that you created the user interface interfere with the results of this method?
- Is this method something you think Medius could perform in the future?
- Are you missing something with this method? What in that case?

This interview and the observations that were made during the evaluations is the foundation of the analysis of the different evaluation methods.

The number of test persons was decided according to Nielsen (2000) and with the time-limit in mind. Nielsen states that the best results come from testing no more than five users. This when running as many small tests that can be afforded.

### 7.2 Cognitive Walkthrough

The goal with performing this cognitive walkthrough was to find potential issues in the user interface as well as review the user interface deeply and really think about the users and their perspective.
The user experience expert walked through the interface and performed the eight tasks that were created for this first evaluation session. These tasks can be found in appendix B. All tests were held in Swedish, and thereby are the scenarios written in Swedish. After every task that were performed these two questions was asked and answered by the user experience expert.

- Will the user know what to do in this step? Is complex problem solving needed to figure out what to do?
- If the user does the right thing, will they know that they did the right thing, and are making progress toward their goal? Is complex problem solving needed to interpret the feedback?

After this walkthrough the evaluation method was evaluated through the interview that was described previously.

While performing this method, performance metrics was collected in form of the time spent on each task. Self-reported metrics was collected from the post-method questions; issues-based metrics was covered by the two post-task questions. The tester in this method was the user experience expert and the author of this thesis acted as an observer in order to monitor the work with this method.

### 7.3 Coaching Method with Interview

The goal of performing the coaching method combined with interview in an early stage of the software development process is to get user feedback on the system early. By analyzing things like; which system-related questions did the testers ask and the user’s ability to correctly complete the task, early in the development helps the interaction designer to understand interface-related issues in the design.

The reason that we combined this method with a small interview at the end was to be able to get more user feedback, both on the method and the system.

A test protocol was created in order to give the same information to all the testers. This test protocol is found in appendix D. A pilot test was conducted in order to make sure that everything was in order. The scenarios that were used in the cognitive walkthrough were the same for the whole first test session, and can be found in appendix B. All the tests were held in Swedish, and thereby are the scenarios and the test protocol written in Swedish. While performing the coaching method there was one coach/observer and one observer/computer. The coach/observer was responsible for getting the testers in the right direction without steering them too much towards the solution. The observer/computer was in charge of the mockup and the behavior of the tester.
During this test the task completion was recorded, as well as time on task. Other metrics such as self-reported was provided by the method itself, and performance metrics were included in the interview. Five test persons conducted this test, excluding the pilot tester, this according to Nielsen (2000). All six testers as well as the coach/observer and the computer/observer are working at Medius at the Linköping office.

7.4 Consistency Inspection

Consistency inspection verifies the consistency of the system, for example fonts, terminology, layouts and so forth. However, to perform an original consistency inspection there should be an evaluation team consisting members from each development team that have the authority to change the design. In Medius case, there are two development teams, but one is located in Poland, so this method in its original was not interesting. We chose to perform this method with a tweak because of the results from the prestudy and the fact that consistency of the system was a high priority. The focus became details in the user interface.

The tweak we made was to have a consistency inspection with the user experience expert together with a colleague that is interested in the user interface. In this case this colleague was the author of this thesis.

7.5 Question-Asking Protocol

The question-asking protocol enables the testers to ask interface related questions to the user while they are solving the tasks provided to them. At the same time the user thinks-aloud so the testers can follow the user’s mental activity regarding the user interface.

As for the coaching method, a test protocol was created in order to give the same information to all the testers. This test protocol is found in appendix E and is written in Swedish, because of the fact that the tests were held in Swedish. Self-reported usability metrics was provided by the method itself, and performance metrics was included in the interview. A pilot test was conducted in order to make sure that everything was in order. The tasks used in this method are the same as some of the tasks from the first test session. These tasks can be found in appendix C, and are also written in Swedish. Five test persons conducted this test, excluding the pilot tester, this according to Nielsen (2000). There were two observers, one of them was the question-asker and the other was the test-leader. All six testers as well as the observers are working at Medius at the Linköping office.
Part IV

Results
Chapter 8

Evaluation Methods

This chapter will present the results of the usability evaluation methods. It will only cover the results regarding the different usability evaluation methods and not the results regarding the user interface. The feedback on the user interface was further considered by the user experience expert at Medius.

8.1 Cognitive Walkthrough

In this section the results from the cognitive walkthrough are presented. The results are divided into two parts, the first one addresses the results of the cognitive walkthrough method with the user experience expert, the second is a SWOT-analysis over the cognitive walkthrough method.

8.1.1 Execution and post execution interview

During the walkthrough there was one observer who observed and measured completion-time, logged the comments and questions from the user experience expert about the user interface in general. The other person that was present was the user experience expert who was the one performing the cognitive walkthrough.

In general the user experience expert, the one who walked through the interface, was pleased with the cognitive walkthrough method. He thought it was easy to perform, and the fact that someone guided the evaluation session was a big help. He, however, expressed concern about the easiness to ”cheat” while performing this method. If the time is lacking and the evaluation needs to be done, one may rush through it and not perform it thoroughly.

Even though the user experience expert who walked through the interface scenario by scenario had designed the user interface, potential interface
issues were discovered. Although the UX expert expressed a worry about the lack of user input, this because of the importance of other people’s opinions. He also expressed a need to evaluate different design proposals and compare them to each other. The cognitive walkthrough method does not cover comparisons of different proposals, but because of the simplicity of the method and the fact that the method only needs one tester makes it a good method to use on different design proposals. Then manually compare and weigh the advantages and disadvantages with the different proposals.

### 8.1.2 SWOT-analysis

<table>
<thead>
<tr>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to perform</td>
<td>Highly dependent on the scenarios</td>
</tr>
<tr>
<td>The evaluation itself is not very time consuming</td>
<td>The UE expert is the one that is evaluating his own work</td>
</tr>
<tr>
<td>Number of test persons is one, therefore easy to fit into a schedule</td>
<td>Only one person’s opinions that is considered</td>
</tr>
<tr>
<td>Can find potential UI-bugs early</td>
<td>Does not include actual users</td>
</tr>
<tr>
<td>Does not require a fully functional mockup</td>
<td></td>
</tr>
<tr>
<td>Cheap to test</td>
<td></td>
</tr>
<tr>
<td>Could be used both early as well as late in the development process</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rethink the user interface before the development has begun</td>
<td>The evaluation only covers the scenarios written</td>
</tr>
<tr>
<td>Higher quality of the usability evaluation if this method is used before another</td>
<td>If the evaluator is lacking time and may stress through the evaluation</td>
</tr>
</tbody>
</table>

### 8.2 Coaching Method and Interview

This section presents the results from the test method called coaching method which was performed together with an interview at the end of the each test session. The results are divided into three parts, execution which describes the results from the coaching method; interviews which contains the results from the interviews conducted and the last part is a SWOT-analysis over the coaching method in combination with interview.

#### 8.2.1 Execution

During the test sessions the comments from the coach and the system-related questions were recorded. This was in order to get as much
feedback on the system as possible. As written previous, the interface results and feedback on the system is not documented in this thesis report. But in the future as well as this time the questions asked by the testers will give the user experience expert a good overlook over the potential and current issues with the user interface.

8.2.2 Interviews

There was two types of interviews held regarding this evaluation method, post coaching interview and post evaluation interview. The post coaching interview was held with every test person and was divided into two parts. The first part was regarding the interface design and the second about the evaluation method and evaluation of the interface in general. This thesis will only address the results from the post coaching interview part II and the post evaluation interview that was held with the user experience expert after all the test sessions.

Post coaching interview part II

The testers generally thought that the coaching concept was a good way of providing feedback, and it is a good and cheap way of testing the basic concept of the system. They liked the idea that the observers could see if there is something in the interface that a lot of users do not understand, and through that improve the user interface. Some of the testers pointed out that the testers learn the system and with the help of the coach they do not get stuck.

A potential concern that was expressed from one of the testers regarding this technique was about the coach. That the test person could get uncomfortable because of the fact that a person who has the answers to all of their questions, are right beside themselves.

Post evaluation interview

The user experience expert was really pleased with this method, and liked the layup of the method. He valued the fact that the coach could observe the tester in a natural way without giving the impression of that the coach is observing. That the method was easy to perform was a big advantage as was the fact that the users will learn the system during the session and get direct answers to their system related questions.

The user experience expert expressed some concern regarding the fact that the coach could ”over-coach” and instead of helping, accidentally inhibit the tester. Another concern was how to gather test persons to future evaluations. How to motivate all employees that usability evaluation is important and in the end everyone gains from usability evaluations.
8.2.3 SWOT-analysis

<table>
<thead>
<tr>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluated the system with actual users of the system</td>
<td>Coach knows everything and the user could be intimidated</td>
</tr>
<tr>
<td>Map system-related questions to the interface</td>
<td>Highly dependent on the scenarios</td>
</tr>
<tr>
<td>Easy to perform</td>
<td></td>
</tr>
<tr>
<td>Cheap to test, especially with mockups</td>
<td></td>
</tr>
<tr>
<td>Enables a discussion about the user interface</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Educate the users in the system during an evaluation</td>
<td>The potential of inhibit the tester and ”over-coach”</td>
</tr>
<tr>
<td>Is suitable both early in the development process as well as late</td>
<td>Could be hard to find suiting test persons</td>
</tr>
</tbody>
</table>

8.3 Consistency Inspection

The objective with this section is to present the results from the consistency inspection. The results are divided into two parts, the first part is about the execution and the post execution interview and the second is a SWOT-analysis made over the evaluation method.

8.3.1 Execution and post execution interview

The performance of this usability evaluation method was tricky according to the user experience expert. He thought that it was hard not to have scenarios to lean back on, otherwise the scenarios created must be very thorough and include every aspect of the user interface. The fact that this method covers a lot of the user interface was, according to the user experience expert, an advantage, but perhaps the different user interface aspects needs to be prioritized so that they are guaranteed to be covered by the evaluation.

The user experience expert liked the fact that the evaluation did not take much time and that the consistency throughout the system is controlled. He also liked to get questioned about the user interface, how his thoughts went when designing it. To be able to discuss the user interface, get valid tips and some comments on it was highly valued by the expert.
### 8.3.2 SWOT-analysis

<table>
<thead>
<tr>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to perform</td>
<td>Highly dependent on the testers</td>
</tr>
<tr>
<td>Makes the interfaces in different parts in the system look cohesive</td>
<td>Needs a usability interested colleague to perform the evaluation with</td>
</tr>
<tr>
<td>Can be used as the finishing touch on the user interface before a release</td>
<td>Easy to get stuck and focus too much on small and insignificant details</td>
</tr>
<tr>
<td>Enables a discussion about the interface</td>
<td></td>
</tr>
<tr>
<td>The evaluation itself is not very time consuming</td>
<td></td>
</tr>
<tr>
<td>Evaluates the entire system</td>
<td></td>
</tr>
<tr>
<td>Cheap to test on the finished system</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>With the right agenda, the feedback could be very useful in the end of the development process</td>
<td>Could be hard to find a colleague that have knowledge as well as the time to spend on this method</td>
</tr>
<tr>
<td>Can result in a more cohesive system</td>
<td>Evaluator is lacking time and therefore just stresses through the evaluation</td>
</tr>
<tr>
<td></td>
<td>Is easy the haste through if time is an issue</td>
</tr>
</tbody>
</table>

### 8.4 Question-Asking Protocol

The aim with this section is to present the results from the question-asking protocol evaluation. The results are divided into three parts. The first part will cover the results from the execution of the evaluation method. The second, interviews, contains the results from the interviews conducted and last the SWOT-analysis which contains a SWOT-analysis over the question-asking protocol.

#### 8.4.1 Execution

During the test sessions the questions from the user experience expert regarding the user interface was recorded, this to be able to get as much feedback on the system as possible. As written previous the interface results and feedback on the system will not be documented in this thesis report.
8.4.2 Interviews

In the question-asking method, two different types of interviews were held. Both were regarding the usability evaluation method, but the questions differed regarding who was the interviewee. The testers were interviewed in a post question-asking interview and the user experience expert in a post evaluation interview.

Post coaching interview

All of the testers thought this method was a good method to provide feedback with, some with emphasis on the importance that the testers really think aloud. They appreciated the fact that they were informed of the fact that questions could emerge and asked during the test session and were not bothered by the questions.

They liked the fact that the observers could see their movement and got feedback on the system; what the users understand and may have trouble with. As well as eventual problems could be caught before the system is released to a customer.

Another comment was; that it was nice to test and get some sort of education in MediusFlow, because as it is today, the employees do not feel like they get any education in the system. Some thought that the tests also should be performed on customers from higher age groups, because of the fact that many that uses MediusFlow belongs in these age groups.

Post evaluation interview

The user experience expert though it was nice to be able to comment and ask questions during the test session. Although he felt that it may be disturbing for the tester if they are asked too many questions, that the formulation of the questions are important as well as the amount, how many and when they are asked. A disadvantage with this method according to the user experience expert was the lack of discussion after the questions. He did want to disturb the tester too much with long and complicated questions and therefore there was never any long and thorough discussions regarding the user interface.

When asking questions during the test session the questions could be leading and help the testers out with the tasks. The questions could even be considered offensive by the tester and make them feel challenged.

A concern with the question-asking protocol was finding test persons to future evaluations. How the user experience expert can find willing colleagues and at the same time convince the company that everyone gains from usability evaluation.
8.4.3 SWOT-analysis

<table>
<thead>
<tr>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluated the system with actual users of the system</td>
<td>May disturb the testers with the questions</td>
</tr>
<tr>
<td>Gets direct feedback on the system while using it</td>
<td>No deep discussion about the user interface</td>
</tr>
<tr>
<td>Easy to perform</td>
<td>Highly dependent on the scenarios</td>
</tr>
<tr>
<td>The evaluation itself is not very time consuming</td>
<td></td>
</tr>
<tr>
<td>Cheap to test on the finished system</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Opportunities</td>
<td>Threats</td>
</tr>
<tr>
<td>Is suitable both early in the development process as well as late</td>
<td>The users could feel challenged by the questions</td>
</tr>
<tr>
<td></td>
<td>The testers could be offended by the questions and get a feeling of inferiority</td>
</tr>
<tr>
<td></td>
<td>Could be hard to find suiting test persons</td>
</tr>
</tbody>
</table>

8.5 General results

A common thing that was consistent through all the testers is the fact that all of the testers expressed that they are willing to participate in similar sessions in the future. This as long as their bosses are approving. Most felt that is was a good idea to use the internal resources in the company to evaluate the user interface and they appreciated that they were given the opportunity to give feedback on the system directly to the user experience expert.

The majority of the evaluators emphasized the importance of testing, and that testing should be a high priority. That the development of MediusFlow will benefit from usability evaluations as well as the whole company. The product would be better and the overall work with the system would be better in form of less support and happier customers.

The different evaluation methods were all simple to perform and did not require much work. This is if the prework is thorough and that it is clearly specifying the tasks from the requirements. This in combination with the wireframes are created specific to these scenarios, the time having to be spent on the evaluation is not much. The methods that involves more testers or evaluators require more time but on the other hand they provide feedback from actual users.
Chapter 9

Evaluation suggestion

This suggestion corresponds to how Medius looks today as well as the needs of Medius according to the prestudy that was made, in combination with the different evaluation methods that were presented in section 4.4.

9.1 Evaluate MediusFlow’s usability

The author’s recommendation is to evaluate the user interface both in the beginning of the development process as well as somewhere near the end of the development, with emphasis on the early evaluation. This would help the developers to develop the final user interface as early as possible. The advantage of this is that the developers would be able to focus more on functional requirements and less on the user interface later in the development. If the user interface changes much in a later stage in the development process the developers can be forced to rethink and rewrite some of the functionality to fit the new user interface which will be a waste of time and money. Early evaluations of the system will hopefully guide the interaction designer to design a good user interface that is tailored to the users.

Because of the fact that making big changes in the user interface later in the development process can be devastating for the system, or even impossible to perform, the author recommends that at least one evaluation should be done before the development of that initiative starts. This evaluation should be done directly after the requirements and the wireframes are created.

Even if the system is evaluated early in the development process, it is important to evaluate the interface later in the process as well. When evaluating early in the development process the evaluation is on a wireframe, often in form of a paper prototype. A paper prototype will not do the finished system enough justice. Things that were considered unclear
on the paper prototype may be perfectly clear in the finished system, and vice versa. Things that are clear in the prototype may not be so in the finished system. When evaluating late in the development process, only small changes are desirable and not huge, game changing ones.

9.2 Evaluation Methods suited for Medius

When it comes to the different usability evaluation methods, the suggestion is based on both the ones that were evaluated and those which were not. This section is divided into the different phases of the software development process they have at Medius. This process can be found in chapter 6, in picture 6.4. An overall view of the suggestion is found in figure 9.1

9.2.1 Preparation phase

Because of the fact that one only want the big user interface changes in the beginning of the development process, it is important to choose evaluation method and corresponding metrics accordingly. My suggestion is to perform a cognitive walkthrough followed by an evaluation involving users, such as:

- Coaching Method
- Thinking Aloud Method
- Question-Asking Protocol

As seen in previous chapter, the question-assembling protocol has the most severe threat. No-one can afford to make the tester’s feel challenged and offended. This method is more suitable to use when the test persons are from Medius, but if it is a first time tester, they may not be willing to participate in another evaluation session.

In this case I would recommend the coaching method in combination with the thinking aloud method and with an interview afterward. When combining these together, the result is more feedback from the users. This in form of the system related questions as well as the users’ thoughts during the session, and a discussion as well as the opportunity to ask the questions to the users. Why they thought like that, what they thought about this and so forth.

9.2.2 Hardening phase

When in the hardening phase the development are mostly done, and in this phase the changes to the user interface are preferably only small,
cosmetic ones. This phase is the one just before the general availability of that version of MediusFlow. Methods and metrics used in this phase should correspond to the goals of the changes only being small and be of an overviewing type.

Suitable methods are:

- Feature Inspection
- Consistency Inspection
- Coaching Method
- Thinking Aloud Method
- Question-Asking Protocol

In this case I would recommend a second round of coaching method in combination with thinking aloud and interviews afterward as well as a consistency inspection with the tweak we did in the study phase.

All the prework for the coaching method have been done during the preparation phase and it will only be the last stages to perform, such as the evaluations and processing the data from the evaluations.

When it comes to the consistency inspection, it is not feasible to meet up with the developers in Poland as the original method is suggests. A solution to that problem would be to have a video conference, but in my opinion, too much information gets lost and the discussion would not be as good as it would be when talking face to face.

The consistency inspection in the tweaked form could be a good way of involving the front-end programmer more in the design process. It could also be a sort of seal of approval, where the product owner and the user experience expert discusses the final touches to the interface.

9.2.3 Maintenance phase

When the software is done, it would be a good idea to actually see how the software is used by the customers. What they think, what they are missing, if there are any issues with it and then tie everything together. During an evaluation in this stage the evaluation should be on the system as a whole, not only the user interface but the whole user experience. User experience involves for example, ease of use and efficiency of the system.

Suitable methods are:

- Field Observation
- Contextual Inquiry
- User Feedback
In this case I would recommend a time-boxed field observation, because of the fact that the evaluators get to see and observe how the software is used. How long this time-box should be can be decided by the evaluator together with the observed user. In my opinion it is important to time-box and be efficient when evaluating. This evaluation can be used internally in Medius, this because of the fact that they use their own and the newest version of their software.

9.3 Test persons

The author would recommend that the test persons in the evaluations are Medius employees, both because of the convenience; they are easy to get a hold of and are interested in the company. The ten persons that was a part of the evaluations were all positive to participate in future evaluations, so a list of all willing personnel at Medius was started.

In the future, when the evaluation is well proven and has a conspicuous role in the development process, then it is important to involve other users more than Medius’ own personnel. It could be motivated to the bosses that the coaching method in the hardening phase could be looked at as a continuing education in MediusFlow. The personnel will by this coaching be better at MediusFlow and therefore become better salesmen, project leaders and so forth.
Figure 9.1: Suggestion for usability evaluation in Medius' current process
Part V

Completion
Chapter 10

Discussion

This case study started with a prestudy which purpose was to get a good view over Medius as a company and their development process of MediusFlow. Based on this prestudy, four different usability evaluation methods were chosen, Cognitive Walkthrough, Coaching Method, Question-Asking Protocol, and Consistency Inspection. These methods were tested by performing them and evaluated based on the evaluators opinions as well as the user experience experts. With the SWOT-analysis’s as the basis, a suggestion of which usability evaluation methods should be used as well as when in the development process they should be used. The results from the different stages of the usability evaluation process will be further discussed below.

10.1 Usability Evaluations

When investigating which usability evaluation methods that existed and how usability evaluation is used, it was hard to find information about usability evaluation in general. The problem was not only the lack of information but it was also the vagueness of the information there was. The basis of this thesis work is from Ivory (2001) and her phd. dissertation and then elaborated by other sources.

10.2 Performing the test evaluations

During the evaluation of the software it was noticed that it is good to have done a detailed pre-work, this in order to save time when evaluating. The test cases can be prepared when writing the requirements, and have the requirements as a basis to the test cases. At the same time the personas can be updated to fit the new initiative and the functionality that comes along with it. Another advantage with doing a detailed pre-work in the
preparation phase is that the uncertainties in the development of the product will be minimized.

10.2.1 Testing protocols

Whilst testing the different usability evaluations, a test protocol was followed to give all the testers the same information. This protocol is a good way of informing the testers about the evaluation and what is to come. It was shown especially important when conducting the question-asking protocol method. None of the testers felt like they were challenged or offended by the questions. The majority of the testers expressed that it was because of the fact that they were informed about the layup of the test in the introduction of the test.

It is important to think through what to write in the protocol, the formulation of the sentence could be the reason that an evaluation is unsuccessful. For example, there is a big difference between; ”You should work as fast as possible, but be thorough” and ”Try to be time efficient and thorough”. The first sentence could stress the tester into making a lot of mistakes because they want to solve the problem as fast as possible.

10.2.2 Selection of metrics

When performing the test evaluations it is essential to the outcome of the evaluations that it is decided what the desired outcome is. This is regulated both by the evaluation method as well as the usability metrics chosen. If some metric is forgotten or if the wrong metrics are chosen, the result from the evaluation will not match the desired result. For example, if task-completion metric is chosen, it is important to specify when and how a task is defined as complete.

During the selection of metrics it is also important to think about the combination between the usability evaluation method and the usability metric, will they collaborate well together? Take task-completion as example again, combining this metric with the coaching method. Is this metric a valid metric if the coach coaches the evaluator into the right path and the solution to the problem?

10.2.3 Scenario design

Writing the scenarios that the evaluations will be based on, the coverage of these tasks are crucial for the evaluation. If the tasks do not cover a specific part of the user interface, that part will not be evaluated. Although it is better to test a little than nothing at all, the evaluations could return results that do not match the wanted results, or return the right results but on the wrong part of the user interface.
10.2.4 Test persons

As briefly mentioned in section 9.3, the coaching method used in the hardening phase could be used as a continuing education of Medius personnel. It could also be used when educating new users in MediusFlow and thereby get the results of the test for free. The new users would need the time for the education and the user experience expert would get input on the system.

10.3 Reliability and Validity

Reliability and validity are ways of indicating and communicating the accuracy of research processes and the credibility of research findings. Validity is measuring what is relevant in the context and using the right thing at the right time. Reliability on the other hand is questioning if the measuring is done in a trusted way. If we are measuring the wrong thing (low validity), then the measuring is not done in a trusted way, low validity therefore causes low reliability. (Roberts et al., 2006)

10.3.1 Prestudy interviews

I would assess the validity of the prestudy interviews as high, because I got answers to all of my questions in one interview iteration. Because of the fact that I interviewed six Medius employees with different roles at the company and triangulated the data I would estimate the validation to be high. After the interviews, the collected data was summarized and sent back to the interviewees for them to comment and approve.

The validity could be higher; this would be done by interviewing more employees with other roles than the ones interviewed in this study.

Because of the fact that I sent back the summarized version of the interviews, I would assess the reliability to be high. All of the interviewees had a chance to verify and approve the outcome from the interviews.

10.3.2 Performance of the test evaluations

In order to get as high validity as possible from the test evaluations, all of the evaluations that were done with a specific method had the same duration and were conducted at the same place. All of them were given the same information about the test by using a test protocol. The number of testers was chosen partly based on Nielsens article written in 2000, that stated that one only need to test on five users. This article is widely discussed and criticized, but having time as an important factor, the number of testers could not be further increased. The fact that all of the
testers were Medius employees and no other users of MediusFlow makes the validity a little lower, but in general, rather high.

10.4 Suggestion

Because of the fact that only a selection of usability evaluation methods were considered in this thesis, there could be other methods that also would suit Medius and its needs. Because the time was limited, there needed to be some limitations and thereby were some of the UEMs not investigated further.

10.5 Future work

In order to carry on with this work, the possibility and approach of integrating usability evaluation would need to be discussed. More of the methods could be tested in a second testing of different usability evaluation methods; this in order to get a broader span of the different methods that could be used.

Usability evaluations should then be integrated and made an obvious step in the development process. Future evaluations should also be made on other persons than only Medius personnel.

I would like to cite one of the interviewees, ”In about ten years, every user will have grown up with Facebook in the digital age, so the purchasers will be accustomed to excellent technique.” (Translated from Swedish to English by the author). This shows how important it is to think about the usability and the user experience when designing software.
Chapter 11

Conclusion

There is a clear picture that testing is an important part of software development, including the work with user experience. It is the whole system that is being evaluated when a user uses it, not only the functionality or the user interface. It is the system as a whole that gives the users their impression.

In section 1.2 the main research questions were established as:

- How can usability evaluation methods be integrated into an agile development process?
- Which methods can be used and how are they used?

There is no right way of integrating usability evaluations in an agile software development process. The usability evaluation process needs to be customized to the current software development process. Every company has a different development process map that they follow and Medius is no different. In this case the current software development process was a basis to the suggestion made of a possible integration of usability evaluations.

When it comes to which methods can be used the issue is that there are many methods that can be used to evaluate the usability, and any is better than none. All methods are used in different ways (presented in section 4.4) and are best suited in different parts of the development process.

It is intended that this thesis will result in an integration of a usability evaluation process in the current software development process. A suggestion is made of how this can be done and which methods could be suited in the different phases of the software development process.
Bibliography


Appendix A

Personas

Peter Authorizer
Department manager

Motivation and beliefs
I’m a department manager in a medium sized manufacturing company. I’m a very busy person and I want things to run smoothly. I get irritated when the requisitioners don’t fill in the PO requisition correctly and complete.

Knowledge about systems, process and organization is average
I have to authorize all purchases made by employees in my department. If a purchase is more expensive than 100,000 SEK, then my boss will have to approve as well.

Expectations on the procurement system
What I would really like is to just receive an email with information about what needs to be authorized and the as easy as possible to the authorization. I also need to be able to reject the requisition with a comment so that the requisitioner can do it again and correct whatever I rejected.

The only time I want to login in to the system (Medius) is when I want to get an overview (run reports) of what has been purchased in my department etc.
Anna Authorizer

Anna is primarily not working with business systems and has only a basic understanding of accounts payable, coding and authorization procedures.

There are lots of Annas in the organization, making her one of the most common user of Medius Flow.

Anna typically authorizes a couple of invoices on a weekly or sometimes monthly basis. She gets irritated when she can’t remember how to use the application and when it is not automating the next step for her. She gets happy when the application tells her she got work to do and presents it with a good overview that clearly shows her what to do. She is not always sitting at her desk so she wants to be able to be notified and access the screen from any place and any device she happens have access to.

Sometimes invoices needs to be checked with a supplier or a colleague, then it is handy for her to be able to mark it up, put notes on it and have reminder set for when it is time to send a reminder to the other party. This is something she used to handle by putting the invoice on her desk with a post it note on in the old days.

In general Easiness is the most important aspect for her. Easiness can be broken down as shown in the graph below.

![Diagram showing the breakdown of Easiness](image_url)
Appendix B

Mobile Scenarios


**Konteringstabell:**

<table>
<thead>
<tr>
<th>Typ av inköp/Konteras på följande konto:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bilhyra 5820 - Car hire</td>
<td></td>
</tr>
<tr>
<td>Datorutrustning 5462 - Purchase computer clients</td>
<td></td>
</tr>
<tr>
<td>Kontorsmateriell 5410 - Consumable equipment and suppliers</td>
<td></td>
</tr>
<tr>
<td>Tägbiljett 5810 - Tickets</td>
<td></td>
</tr>
<tr>
<td>Kostnadsställe: 30 R&amp;D</td>
<td></td>
</tr>
<tr>
<td>Momsgrupp: 25</td>
<td></td>
</tr>
</tbody>
</table>

74
Appendix C

Expence Invoice Scenarios


Konteringstabell:

Typ av inköp/Konteras på följande konto:
- Bilhyra 5820 - Car hire
- Datorutrustning 5462 - Purchase computer clients
- Kontorsmateriell 5410 - Consumable equipment and suppliers
- Tågbiljett 5810 - Tickets

Kostnadsställe: 30 R&D
Momsgrupp: 25
Appendix D

Test protocol Coaching Method


Det är viktigt att du vet att vi inte testar dig utan metoden och i andra hand systemet. Har du några frågor så långt?

Rollen: Rollen du har är den i form av Peter, han är ansvarig för inköp på R&D avdelningen och får lov att godkänna belopp upp till 100 000 SEK.

Instruktioner:

- Tänk på att försöka lösa scenarionå sa tidseffektivt och noggrant som möjligt
- Tänk på att du gärna får fråga frågor om systemet under tiden
- Läs scenariot högt så vet vi när du börjar

Är allt klart så startar vi där.
Appendix E

Test protocol Question Asking Protocol


Metoden vi kommer använda oss av idag är en testmetod som heter ”Question-Asking Protocol” och är en version av metoden ”Think-Aloud”. I denna metod ska du tänka högt, det vill säga att du ska säga vad du tänker och att vi som testledare får frågor om interfacet under testets gång. Det är för att vi ska kunna få en bättre idé om dina tankar om interfacet. Det vi ska testa idag är webb-applikationen med fokus på varufaktura modulen som är i ”continuous improvement” fasen, det vill säga att den är fullt implementerad och ständigt utvecklas för att bli bättre.

Det är viktigt att du vet att vi inte testar dig utan metoden och i andra hand systemet. Har du några frågor så långt?

Rollen: Rollen du har är den i form av Anna, hon är ansvarig för inköp på R&D avdelningen och får lov att godkänna belopp upp till 100 000 SEK.

Instruktioner:

- Tänk på att försöka lösa scenariena så tidseffektivt och noggrant som möjligt
- Tänk på att du gärna får frågor om systemet under tiden
- Läs scenariot högt så vet vi när du börjar

Är allt klart så startar vi där.
Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår. Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmanens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art. Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydda mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart. För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a considerable time from the date of publication barring exceptional circumstances. The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Malin Neveryd