DISSERTATIONS IN
FORESTRY AND
NATURAL SCIENCES

Susanna Martikainen

Towards Better Usability

Usability and End-User Participation in Healthcare Information Technology Systems Development

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences No 201

UNIVERSITY OF
EASTERN FINLAND

SUSANNA MARTIKAINEN

Towards Better Usability

Usability and End-User Participation in Healthcare Information Technology Systems Development

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 201

Academic Dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in Auditorium TTA in the Tietoteknia Building at the University of Eastern Finland, Kuopio, on December 18th, 2015 at 12 o'clock noon.
School of Computing

Grano
Jyväskylä, 2015
Editors: Research Dir. Pertti Pasanen,
Profs. Pekka Kilpeläinen, Kai Peiponen, and Matti Vornanen

Distribution:
University of Eastern Finland Library / Sales of publications
P.O.Box 107, FI-80101 Joensuu, Finland
tel. +358-50-3058396
http//www.uef.fi/kirjasto

ISBN: 978-952-61-1980-9 (nid.)
ISSNL: 1798-5668
ISSN: 1798-5668

ISBN: 978-952-61-1981-6 (PDF)
ISSNL: 1798-5668
ISSN: 1798-5676

Author's address University of Eastern Finland
School of Computing
P.O. Box 1627
70211 KUOPIO
FINLAND
email: susanna.martikainen@dnainternet.net

Supervisors Docent Mikko Korpela, DTech.
University of Eastern Finland
School of Computing
P.O. Box 1627
70211 KUOPIO
FINLAND
email: mikko.korpela@uef.fi

Vice Dean, Professor Satu Miettinen, PhD
University of Lapland
The Faculty of Art and Design
Department of Industrial Design
P.O.Box 122
96101 ROVANIEMI
FINLAND
email: satu.miettinen@ulapland.fi

Professor Markku Tukiainen, PhD
University of Eastern Finland
School of Computing
P.O.Box 111
80101 Joensuu
FINLAND
email: markku.tukiainen@uef.fi

Reviewers

Adjunct Professor Yvonne Dittrich
IT University of Copenhagen
Software Development Group
Rued Langgaards Vej 7
DK-2300 COPENHAGEN S
DENMARK
email: ydi@itu.dk

Professor Samuli Pekkola, PhD
Tampere University of Technology
Department of Information Management and Logistics
PO Box 541
33101 TAMPERE
FINLAND
email: samuli.pekkola@tut.fi

Opponent

Professor Marko Nieminen, D.Sc.(Tech.)
Aalto University
School of Science
Department of Computer Science and Engineering Software
Business and Engineering
Usability and User Interfaces
P.O. Box 15400
00076 Aalto, ESPOO
FINLAND
email: marko.nieminen@aalto.fi

ABSTRACT

Generally, the users of healthcare information systems are not satisfied with the usability of the systems. User-centered design (UCD) methods and user participation are often offered as a solution to the software's usability problems. In the field of health informatics, some studies of user centred development of healthcare IT systems are conducted, and the results are positive. Based on the literature review, UCD methods should be an explicitly standard part of software development.

The main research interest of the present study was to examine what factors impact the usability of Healthcare IT Systems (HITS) development in Finland to understand the reasons for poor usability. This study was conducted with a practical orientation – research questions were raised from practical concerns and from the needs to understand and have better tools for solving practical problems in software development work.

Methods used in this research were action research, questionnaire, and interview.

This study began by experimenting with user centred and participatory methods in healthcare IT system development. Because of the very positive feedback of the UCD methods used and the lack of knowledge on the methods in user participation in HITS development, a need to know users' opinions and experiences with HITS development in a wider scale was raised.

The study continued by discovering end-users' opinions about and experiences on participation in the development and use of healthcare IT systems. Next, how the developers thought about end-user participation and the present state of HITS development was examined. Lastly, a broader landscape view on the present state of HITS development in Finland was studied.

Based on all of those findings and the literature review, 17 problematic areas and 9 recommendations were created for HITS development. Also, further development for an activity-driven information system development (ADISD) model was done.

One of the findings is that user participation is not the only issue affecting the usability of HITS. Studying and modelling the landscape view of HITS development revealed that a number of stakeholders and factors caused challenges for the usability development. By the recommendations and the enriched model, this thesis contributes to the objective of progressing towards better usability of healthcare IT systems.

This thesis consists of four research papers and a summary part.

Universal Decimal Classification: 004.05, 004.41

National Library of Medicine Classification: W 26.5, W 26.55.I4

INSPEC Thesaurus: medical information systems; medical computing; information technology; health care; software engineering; software process improvement; user centred design; information needs; problem solving

Yleinen suomalainen asiasanasto: tietojärjestelmät; tiedonhallinta-järjestelmät; ohjelmistokehitys; tietotekniikka; terveydenhuolto; käyttäjäkeskeinen suunnittelu; käytettävyys; tarpeet; tarvekartoitus; ongelmanratkaisu; mielipiteet; osallistuminen; osallistuva suunnittelu; loppukäyttäjät; Suomi

Acknowledgements

This thesis project was quite a journey. I do not exactly remember how I first encountered the world of health informatics, but this world has been very exciting and interesting from the beginning. A passionate interest in designing user-centred software started when I began to study graphical design. Combining health informatics and user-centred design has been extremely interesting. Without having great people around me from the very beginning of this thesis journey, it would not have been possible to conduct this research.

I want to thank my main supervisor, Dr. Mikko Korpela, for continuous support, excellent guidance, and understanding throughout this journey. Also, I want to thank my other supervisors, Dr. Satu Miettinen and Dr. Markku Tukiainen, for all their support and guidance. Prof. Samuli Pekkola and Prof. Yvonne Dittrich kindly accepted roles as reviewers and, with their guidance and valuable comments, this thesis moved to next level; thank you both. Prof. Marko Nieminen, thank you for accepting role as an opponent.

This study started in ZipIT project at the University of Eastern Finland, and I was able to continue the research during my work at a software company, mostly in my free time. Without the possibility to work in real life software development, this thesis would be entirely different.

To my dearest research sisters, some of them also co-authors of my papers: Pauliina Ikävalko, Tuija Tiihonen, Marika Pentikäinen, Irmeli Luukkonen and Vilma Vainikainen, I owe you so much! I also want to thank the other co-authors of my papers, Johanna Kaipio and Tinja Lääveri. Special thanks also to the ZipIt project people, especially Ritva Silvennoinen.

I want to thank my colleagues for many interesting discussions and support for this research, especially to Hannu Puurunen and Kati Räätäri for giving me the opportunity to work in this such a fascinating business. Soile, Riitta, Asta, Marko, Auli, Petteri, Tuula, Tanja, Perttu, Anssi, Janne, Jukka, Karsu, Simon, Teemu Sami, Mata, Mari, and Heidi - thank you all for interesting discussions concerning health IT system usability development and for giving me opportunities to learn more about this exciting field of healthcare IT systems! Marja, Outi, Minna, and Kati – thank you for your support. Thanks also to all other colleagues who have supported me, too.

I want to thank my friends for giving me something else to think about than this thesis project. Especially: Mari and Janne. Sanna. Riikka and Ville. Jutta and Petri. Anniina.

My family earns the deepest thanks. To my mother Seija and my father Matti: you have always supported and helped me whenever I needed it. Also to my little sisters Eeva and Riikka, you are the best sisters ever. To my sisters’ spouses Jonne and Jussi and my little nieces Vilina and Selina, thank you too! My mother-in-law and father-in-law Maija-Liisa and Matti – thank you for your support! I think you all might have sometimes had second thoughts about my new “projects”, like having another dog, building a new house, collecting old tiles for garden projects, but you all have always been there for me.

I want to thank also my dogs, those deceased and 'Sissi', for taking me to woods to have a break from all the responsibilities, household chores, and studies.

Tero, my husband, without your love and understanding this thesis project would not exist. Thank you also for being both dad and mom to our kids during last months. I love you.

Finally, last but not least, to the most important persons in my life, my children, Minea, Väinö, and Eino: sorry for being such a terrible mother during these last months, because of this thesis, but in the future I promise to be more patient and present for you! Thank you kids, just for being as lovely as you always are, you make my life so vivid!

I also want to thank ISD4D project funded by the Academy of Finland, grant no. 253275.

Kuopio, 18 November 2015

Susanna Martikainen

LIST OF ORIGINAL PUBLICATIONS

This thesis is based on data presented in the following articles, referred to by the Roman numerals I-IV. The publications have been included at the end of the printed version of this thesis with permission by their copyright holders.

I. Martikainen, S., Ikävalko, P., & Korpela, M. (2010). Participatory interaction design in user requirements specification in healthcare. Studies in Health Technology and Informatics, 160(Pt 1), 304-308.

II. Martikainen, S., Viitanen, J., Korpela, M., & Lääveri, T. (2012). Physicians' experiences of participation in healthcare IT development in Finland: Willing but not able. International Journal of Medical Informatics, 81(2), 98-113.

III. Martikainen, S., Korpela, M., & Tiihonen, T. (2014). User participation in healthcare IT development: A developers' viewpoint in Finland. International Journal of Medical Informatics, 83(3), 189-200.

IV. Martikainen, S., Korpela, M., Luukkonen, I., & Vainikainen, V. (2015). Where does the interaction break down? The stakeholder map of health IT systems development and use in Finland. To appear in Finnish Journal of eHealth and eWelfare, 7(4)

AUTHOR'S CONTRIBUTION TO THE JOINT PAPERS

The author of this thesis designed the study and made a significant contribution and was a corresponding author to all the papers.

Paper I. Article 1 was written in close collaboration with Pauliina Ikävalko and Mikko Korpela. The research case study reported in the article was carried out by these authors together. The author of this thesis was the main author and was mainly responsible for writing the sections of introduction, materials, and methods. Pauliina Ikävalko was mainly responsible for writing the results and discussion, and presented the results in the MEDINFO 2010 conference. Mikko Korpela is the primary supervisor of both other authors, and he particularly contributed to structuring the paper and the theoretical background and discussion sections.

Paper II. This questionnaire study was part of a national questionnaire project in spring 2010. The original research idea and the questions on user-oriented and participatory IT development in the questionnaire were defined by the author of this thesis, who was the main author and mainly responsible for the study design and for writing the discussion and conclusion. She analysed the qualitative data in close cooperation with Johanna Kaipio. The introduction, study methods, and statistical analysis were mainly Kaipio's responsibly. Mikko Korpela contributed to structuring the paper and participated in writing the results and conclusions. Tinja Lääveri contributed to the results and was one of the organisers of the large questionnaire.

Paper III. This questionnaire study was designed and carried out and the data was analysed by the author of this thesis. She was the main author and wrote the initial version of the entire paper. Tuija Tiihonen contributed by commenting on the whole paper, analysing the data, and conducting an analysis of context levels. Mikko Korpela mentored the study and participated in designing the questionnaire and writing the paper.

Paper IV. This paper was an interview study and was designed, carried out, analysed, and documented by the author of this thesis. Mikko Korpela participated in the design and contributed by presenting the landscape depicting methodology and also by copy-editing the final version. Irmeli Luukkonen and Vilma Vainikainen contributed to analysing the data and writing the results and discussion.

LIST OF ABBREVIATIONS

ActAD	Activity Analysis and Development
ADISD	Activity-Driven Information System Development
CMM	Capability Maturity Model
DCS	Design Contribution Square
EMR	Electronic medical record
EPR	Electronic patient record
GDD	Goal Directed Design
GUIDe	Goals – User Interface Design – Implementation
HCI	Human-computer interaction
HIS	Healthcare information system
HIT	Healthcare information technology
HITS	Healthcare information technology system
IS	Information Systems (discipline)
ISD	Information system development
ISO / IEC	International Organization for Standardization
IT	Information technology
IxD	Interaction design
MDDD	Medical Device Design and Development
MSCUI	Microsoft Health Common User Interface
PD	Participatory Design
QIP	Quality Improvement Paradigm
QSM	Quality Management System
RQ	Research question
SDLM	Software Development Lifecycle Methodology
SE	Software engineering (discipline and practice)
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability dEtermination
SRQ	Sub-research question
UCD	User-Centred Design
UI	User interface
UX	User experience
WHO	World Health Organization

CONTENTS

1 Introduction

Information technology (IT) has developed into a valuable tool for work activities in healthcare organisations. Recently the focus has increasingly been on design and ergonomics. Usability and user experience factors are now more often the issues that are most significant when assessing the success of any IT project or product.

It has been argued that human-centred design of healthcare IT is today more necessary than ever (Jones, 2013). It has also been stated that usability is possibly a major obstacle or the most important factor hindering widespread adoption of electronic medical records EMRs (Belden, Grayson, & Barnes, 2009; Yen & Bakken, 2012).

1.1 BACKGROUND AND MOTIVATION OF THE RESEARCH

User satisfaction with healthcare software in Finland is poor (e.g. Lääveri, 2008a; Vainionmäki et al., 2008). The last decade has witnessed considerable numbers of articles on poor usability, poor quality, and dissatisfaction with the healthcare software in end-users' media, like in the journal of the Finnish Medical Society (e.g. Jokela, 2008, 2011; Korhonen & Hartikainen, 2009; Korpimies, 2015; Lääveri, 2008b, 2008c; Marin, 2006; Ora J., 2007). Finnish physicians conducted a small-scale user satisfaction and usability study of the most commonly used electronic patient record (EPR) systems in Finland in 2008 (Lääveri, 2008d). The results of the study were discouraging.

User dissatisfaction in healthcare software is also known to exist worldwide (Nielsen, 2005; Wehlou, 2014). There have even been allegations that patients have died because of poor software usability (Eronen, 2007; Jones, 2013). Poor usability can lead to a situation where necessary information is not available when needed, and if necessary information is missing in a care situation, it could be life threatening to patients. *"Usability can sometimes mean the difference between life and death"* (Tullis & Albert, 2008). Miller & Sim (2004) have studied physicians' use of electronic medical records (EMRs) and claim that usability problems caused the users to have to spend extra time to learn to use EMR effectively, and this brings about barriers to achieving benefits and quality improvements with EMRs (Miller & Sim, 2004).

Poor usability of healthcare software is certainly regrettably common. In addition, the results of a recently published doctoral study also indicate that healthcare software does not support clinical tasks well enough and the systems used have multiple usability and interaction related problems (Kaipio, 2011). The software that should be helpful to the healthcare professionals does almost the opposite – it is time consuming and gives not enough value to everyday work. Users use software because it is mandated; however, the software fails in making their professional lives easier.

In addition to the software's poor usability, challenges to the healthcare system are caused by the aging population and rising incidence of chronic health problems in the western world (e.g., Dearden et al 2010). This results in an economic burden to society, because there are fewer working-age people to pay the cost of the treatments.

Healthcare information technology (HIT) has been presented as necessary to enhance and improve the healthcare delivery system (R. Miettinen et al 2003). Sittig & Singh (2011) have presented risks for adverse event, patient harm, and safety hazards, and one of risks can be considered to be usability-related (HIT is used incorrectly by someone). Usability has an important role when assessing if HIT enhances and improves healthcare or not. It has been noted that poor usability slows down physicians' work; for example, in primary care outpatient clinics, less patients can be treated in a day (Kaipio, 2011 p.73).

The ongoing big change, the "digital revolution / digitalization" affects healthcare also; services to citizens are being digitalised. For citizens, too, software usability plays an important role. If usability is poor, services will not be used by citizens and it will further burden healthcare professionals.

1.2 DEFINITIONS OF KEY CONCEPTS

Before moving further to investigate the topic of the thesis, the definitions of the key concepts, as understood in this study, are provided to orient the reading from the beginning. The argumentation around the concepts is provided later in Chapter 2.

1.2.1 Healthcare information technology system (HITS)

This study has been conducted within a research group that understood Information Systems (IS) as social science of information and technology. In that tradition, the term information system means "*the processes of managing (creating, using, storing, exchanging, etc.) information in an organizational setting (in work activities) for a purpose*" – "*a socio-technical entity in the user organization consisting of people (actors), information (contents), and technology (means), linked together by a process directed toward a purpose*" (Luukkonen et al., 2013, p. 449-450).

In day to day language and in other disciplines in computer sciences, too, however, the term information system is often used to refer to a multi-user software system; the core technological part within a socio-technical information system.

To avoid confusion, in this study since Paper III the term **healthcare IT system (HITS)** has been used to refer to a software product or an integration of software products as well as the necessary hardware on which the software runs, used in healthcare organisations such as hospitals and clinics (Paper III, p. 190; cf. Paper II, p. 100). For more discussion see Paper III, sections 2.1 and 2.2.

1.2.2 Usability

The term **usability** is used in this study in a broad meaning, according to the ISO 9241-11 (ISO, 2009) standard definition: *"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"* (Jokela et al., 2003; Paper I, p. 304).

The term is thus used here in a wider meaning than in the well-known definition by Nielsen (2012), as a synonym to what Nielsen calls usefulness. Fundamentally, usability is about supporting users' work (Jokela, 2015). Further discussion is presented in section 2.4.

1.2.3 User participation

User participation is defined by Barki & Hartwick (1989) as "*a set of behaviors or activities performed by users in the system development process*". User participation is recommended "*when referring to the assignments, activities, and behaviors that users or their representatives perform during the system development process*" (Barki & Hartwick, 1994, p. 60).

The concept of user participation is considered and presented in all sub-studies (Paper I, p. 304; Paper II, section 2; Paper III, section 1.3; Paper IV, p. 6-7). Further discussion is provided in section 2.5, including discussion on differences between the concepts of 'user participation' and 'user involvement'.

1.3 RESEARCH PROBLEM, OBJECTIVE, AND QUESTIONS

The poor usability of healthcare IT systems poses problems to healthcare professionals in everyday life. Of course, not all HITS are weak in usability, and understanding what has been taken into account already helps one to learn how to build excellent usability into HITS for healthcare professionals. *How to develop high-usability healthcare software* is the key problem area to which this research aims to contribute.

It is commonly assumed that user-centred design (UCD) methods in system development (also called human-centred design or usability engineering; Jokela et al., 2003) produces better solutions according to the users' point of view. Thus, it seems very obvious to assume that user-centred methods are used to achieve good results in software usability. Jokela et al. argue as follows, relying on a variety of sources:

> *"To improve the usability of software and information systems, the paradigm of user-centered design, UCD, has been proposed by a number of method and methodology books, starting from Nielsen (Jakob Nielsen, 1993) to ones published in late 90's, (Hix & Hartson, 1993), (H. Beyer & Holtzblatt, 1998), (A. Cooper & Saffo, 1999), (Mayhew, 1999) and ending up with a set of very recent ones, (Rosson & Carroll, 2002) and (Vredenburg, Isensee, & Righi, 2002)."* (Jokela et al., 2003, p. 53)

Are user-centred design methods commonly used in healthcare IT development? One answer for that was achieved in a study about medical device design and development: *"Only one out of 11 manufacturers claimed to regularly use formal user centred design methods within the MDDD (medical device design and development) process"* (Money et al., 2011, p. 7). Also in sub-study 1, the present researchers found common methods used in HITS development in Finland at that time (Paper I). The most common method seemed to be user participation in workshops; no other UCD or participatory design (PD) methods were very commonly used at that time.

Do user-centred methods fit in the development of healthcare software? According to Thursky & Mahemoff (2007) and Chan et al. (2011) user-centred design methods are effective when designing healthcare IT systems. According to Jones (2013), though, traditional user-centred design practices are inadequate for solving problems in complex healthcare. Jones also reminds readers that it is easy to design a perfect product or service to "our users", but still remain disconnected from other related stakeholders and systems (Jones, 2013). Healthcare IT systems are complex and consist of a number of software products even within a single organisation.

On this basis, the **research problem** is the dissatisfactory level of the usability of HITS and user participation in HITS development. The aim of this study, the **research objective**, is to find ways of improving the usability of HITS and user participation in HITS development in Finland.

The overall research question is:

RQ: What factors impact the usability of HITS and user participation in HITS development in Finland?

The overall research question is divided into five sub-research questions (SRQ), presented in Table 1.

Table 1: Sub-research questions and expected outcomes.

Sub-research question		Expected outcome
SRQ1	Are user-centred methods useful in HITS development?	Preliminary understanding
SRQ2	What is the current state of the usability of HITS and user participation in HITS development from the users' viewpoint?	Issues in the current state
SRQ3	What is the current state of the usability of HITS and user participation in HITS development from the developers' viewpoint?	Issues in the current state
SRQ4	Which issues in addition to user participation affect the usability development of HITS?	Landscape view of HITS development
SRQ5	What can be done to improve the usability of HITS and user participation in HITS development?	Proposals to development

The purpose of sub-research question 1 is to acquire preliminary understanding of the phenomenon of research interest – the potential usefulness of user-centred methods in HITS development. The understanding generated shall then be used in studying the current state of the research topic from end-users' viewpoint (SRQ2) and from developers' viewpoint (SRQ3).

The purpose of sub-research question 4 is to broaden the view to identify more stakeholders and issues than SRQ2 and SRQ3 will reveal. The first four sub-research questions taken together cover the overall research question (what factors are there that impact the topic), while the purpose of sub-research question 5 is to move towards the research objective of finding ways to improve the current situation.

1.4 STRUCTURE OF THE THESIS

This thesis is organised as follows: Chapter 2 focuses on the theoretical background of this research, with respect to the research questions. The chapter is divided into five parts: health informatics, software development, usability, user participation, and, in the last section of theoretical background, 'usability and user participation in HITS development' is summarised. Chapter 3 describes the research methods used and the research approach chosen. Summaries of each original publication (Paper I-IV) are presented in Chapter 4, while the original research articles are included at the end of this thesis. Chapter 5 analyses the findings from the papers, makes recommendations based on the findings, and presents one model to HITS development. The study as a whole is discussed in Chapter 6. Finally, Chapter 7 concludes this research work.

2 Theoretical Background

This chapter presents the theoretical background of this study. The discipline of this research is health informatics and, more specifically, usability and user participation in healthcare software development (Figure 1).

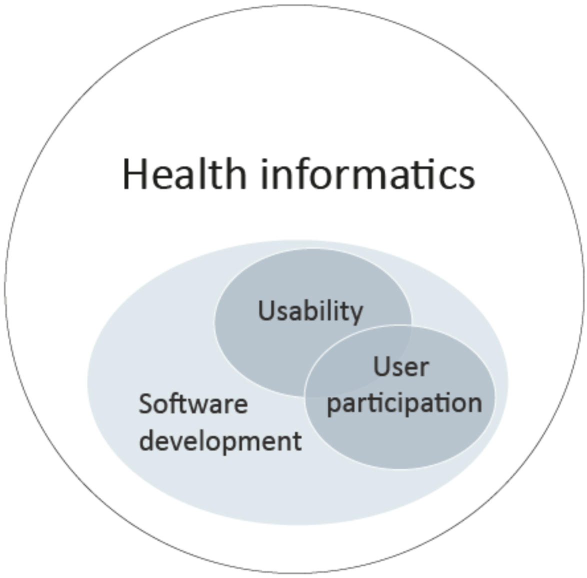


Figure 1. The theoretical background

This chapter is organised as follows: The first two sections are brief overviews of the study's points of departure: section 2.1 is on health informatics and on the specificity of healthcare IT systems, and section 2.2 provides an overview of software development activities and presents agile development in more detail. Also in this section, a view of developing software is enlarged with an activity-driven information system development model (ADISD). Section 2.3 focuses on 'Usability': the user-centred design, practical UCD models, and the integration of agile and UCD approaches. The concept of user participation is investigated in section 2.4. Section 2.5 focuses on summarizing UCD and user participation in health informatics.

2.1 HEALTH INFORMATICS

Merriam-Webster defines **healthcare** as *"the maintaining and restoration of health by the treatment and prevention of disease especially by trained and licensed professionals (as in* *medicine, dentistry, clinical psychology, and public health)"* (Merriam-Webster, 2005). Healthcare is complex work done by educated and licensed professionals. **Health informatics** is defined by Saranto & Korpela (1999, p. 19) as *"the application of information and communication technologies (ICTs) in healthcare (as a field of the science and as a practice)*". Health informatics is also called medical informatics. The Healthcare Information and Management Systems Society (HIMMS) defines medical informatics as "*the interdisciplinary study of the design, development, adoption and application of IT-based innovations in healthcare services delivery, management and planning"* (HIMSS, 2015).

2.1.1 Healthcare in general and in Finland

According to the World Health Organisation (WHO), good healthcare systems deliver *"quality services to all people, when and where they need them"* (WHO, 2015). Different countries have different configurations of healthcare services. According to WHO, healthcare delivery requires "a *robust financing mechanism; a well-trained and adequately paid workforce; reliable information on which to base decisions and policies; well maintained facilities and logistics to deliver quality medicines and technologies"* (WHO, 2015).

As stated in Paper IV, chapter 1.1, the healthcare delivery system in Finland greatly differs in some fundamental aspects from the healthcare systems in countries like USA, United Kingdom, Germany, Denmark, etc. The healthcare delivery system in Finland is described in more detail in Paper II, chapter 1.2. Compared to that description, today there are less municipalities as a result of mergers, and only time will tell how the planned national healthcare reform will change the landscape of healthcare delivery. As stated in papers II and IV, healthcare IT systems were introduced in Finland in the early 1980s and currently HITS are used comprehensively on all levels of healthcare (Reponen, Winblad, & Hämäläinen, 2008; Saranto & Korpela, 1999; Winblad, Reponen, Hämäläinen, & Kangas, 2008). As already stated in Paper II, the main challenges for healthcare IT development in Finland derive from the strongly decentralised healthcare delivery system, a wide diversity of disintegrated software systems in use, as well as the distributed nature of IT systems development.

Like healthcare, the healthcare information systems are important to everyone, not only to those who utilise those systems in their work. The information stored in those systems is about our health and wellness, a matter of life and death.

2.1.2 The specificity of health informatics and healthcare IT systems

As seen in the first paper of this research (Paper I), a difference of developing healthcare information technology compared to other information technology products (e.g. IT products for consumers) was observed. Literature supports the proposition of health informatics being "kind of special" in the IT development sector. Reasons for the specificity are explored in Paper IV, section 1.2, and in this section of the thesis.

Healthcare information systems (HIS) are information systems used for managing health-related information. According to Ge, Paige, & McDermid (2009, p. 53), *"Healthcare information systems are amongst the most complex IT systems in the world."*

Designing software with good usability for knowledge workers such as healthcare workers is a challenge (Miller & Sim, 2004). The use context of healthcare information technology is highly complex, and physicians' and other healthcare professionals' workflows are difficult to understand without medical or healthcare education. Healthcare professionals' work is largely communication and cooperation among different professionals, which needs to be supported by information systems. Information should be available 24/7, which brings about challenges to IT systems also. *"You have to build a plane and fly it at the same time"* – that is, it is challenging to implement new functionalities into use while the system is in continuous use at the same time (Mäkelä, 2006). It is necessary to understand work tasks in order to design software with good usability and user experience. The healthcare work activities where the information systems will be used should be an obvious starting point for development (Nykänen & Karimaa, 2006).

Jones (2013) argues that the design needs of healthcare organisations are probably too domain-specific for the most generalised user experience designers. Usability work with electronic medical records (EMRs) is not straightforward; clinicians use foreign terminology, and because medicine has a lot of specialties, every specialist may have different needs when using EMRs (Smelcer, Miller-Jacobs, & Kantrovich, 2009).

Far more rigorous design and development methods are required in the healthcare environment compared to other information system development (Jones, 2013). User-centred design principles applied throughout the design process provide quality and acceptance among the end-users of healthcare IT systems (Christensen et al 2009; Johnson et al 2005). On the other hand, it has been argued that it is difficult to apply normally used usability methods, e.g. observations, in healthcare in actual patient care situations, because of the confidentiality of patient records and privacy regulations (Smelcer et al., 2009).

Since healthcare is a very information-intensive industry, a large amount of time is spent in information processing tasks (Mykkänen, 2007). Therefore, visual design and information visualisation of the user interfaces of a healthcare IT system are demanding to designers. The information needed in patient care must be displayed in the right situation in the right manner, or it may lead to inefficient care (Johnson et al., 2005). User interface design should have a minimalistic design style, according to Jones: *"to ensure that content is observed precisely and can be acted upon quickly and unambiguously"* (Jones, 2013, p. 216). A system should only display information that is relevant for analysis and decision-making (Mykkänen, 2007).

In the healthcare sector, end-users are not the ones who purchase products or even decide on what to purchase (Miettinen et al., 2003), which can cause problems, especially when developing the usability aspects of products.

Also, the new products are usually only one part of the larger (hospital) IS infrastructure, not "independent products", which brings about challenges to users, especially regarding overall usability. Harrison et al. (2007) argue that the interplay between current socio-technical conditions and new technologies may be overlooked when it is assumed that a HITS will fulfil the software vendor's promises. Challenges to development are also brought about by the legislation and regulations on information management and on the information that must be documented in healthcare activities.

The development and deployment of new HITS products is a long process – one may think that it is a slow process – and in addition to that, "*healthcare facilities are not early adopters*" (Jones, 2013).

Furthermore, Miettinen et al. (2003) have presented a list of problems in the relationship between health technology providers and users (translations from Finnish by this author):

- The way in which healthcare is organised, which is complex and difficult to understand by outsiders (e.g., the user of technology, the person making decisions on acquisition, and the payer can be located in different organisations);
- The difficulties of learning and dialogue, both among healthcare actors and in cooperation with companies;
- Product developers' technology-centredness and inadequate understanding of the requirements of clinical operational use;
- The differences in activities, time spans, and commitment between corporations and public healthcare organisations.

2.2 SOFTWARE DEVELOPMENT

Software development is demanding.

> *"The creation of quality software on time and within budget has been a major problem facing the software industry for several decades."*
>
> (Thayer, Dorfman, & Hunter, 2001)

Multiple skills and techniques are needed to build software that fulfils users' needs. Working habits, models, and methods vary within and between companies. A number of software development process models, also called software lifecycle models, have been developed over the years. There are two types of software lifecycle models: descriptive and prescriptive. According to Scacchi, a descriptive model describes *"the history of how a particular software system was developed"* and a prescriptive model prescribes *"how a new software system should be developed"* (Scacchi, 2001). In this thesis the focus is on prescriptive models.

It is important to use models to support software development; software development models are supposed to benefit the software development. Software engineering (SE) models bring about a systematic and plannable approach to development (Nebe, Zimmermann, & Paelke, 2008). Models also have an important contribution to the overall success of software applications. According to García-Mirales et al. (2013) "*… the quality of software products depends, to a great extent, on the processes used for their development and/or maintenance*".

Development models and methods have a high impact on software products, and thus, on end-users' experience. An understanding of the theory of software development is thus needed in this research before it is possible to contribute to developing high-usability healthcare software. Section 2.2.1 presents an overview of software development activities, and section 2.2.2 summarises what the Software Process Improvement movement can provide to this study. Currently, agile software development methods are mainstream in software development (McInerney & Maurer, 2005). Thus, section 2.2.3 focuses on the agile methodology, particularly on one specific type of agile development: Scrum. Section 2.2.5 broadens the perspective from software development to whole information system development, presenting one model for IS development.

2.2.1 Software development processes

The International Standard *ISO/IEC 12207:2008: System and Software engineering – Software life cycle processes* establishes a common framework for software life cycle processes. The standard can be applied to the acquisition, supply, development, operation, maintenance, and disposal of software. The Process Reference Model presented in the standard does not introduce "*a particular process implementation approach nor does it prescribe a system/software life cycle model, methodology or technique*". Therefore, every organisation needs to produce its own model, for example, for the development activities, using the reference model as a basis (International Standard ISO / IEC 2007, 2008). Figure 2 draws together the grouped life cycle processes presented in the standard.

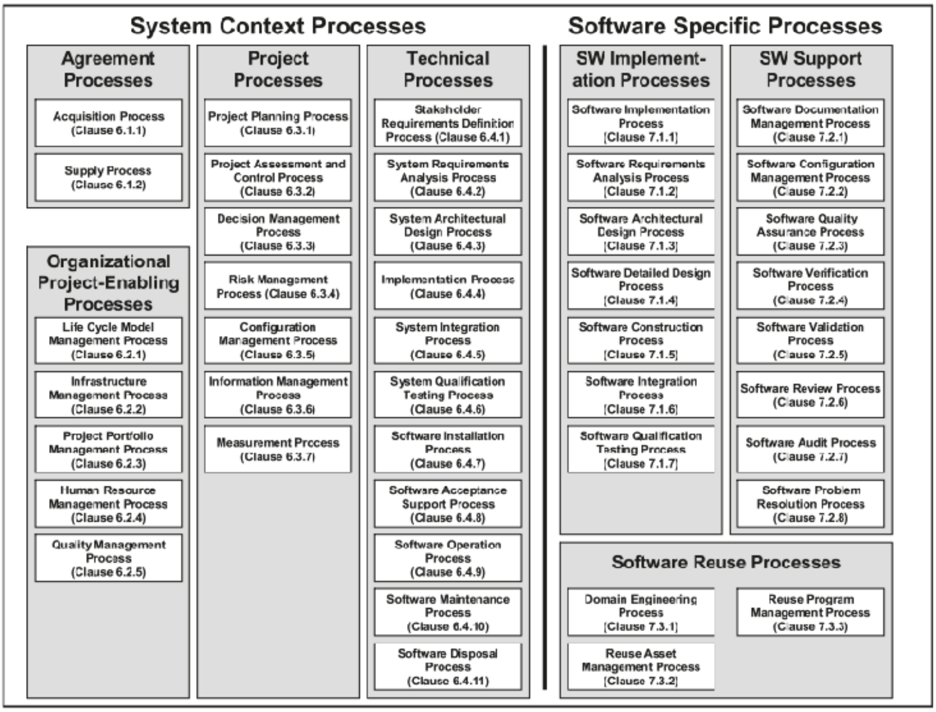


Figure 2. Life Cycle Process groups (International Standard ISO / IEC 2007, 2008)

Sommerville (2007) presents four fundamental activities of the software process: software specification, software design and implementation, software validation, and software evolution. According to Haikala & Märijärvi (2002), the activities of software development are: feasibility study (i.e. preliminary analysis), requirements analysis (i.e. requirements specification, system analysis), design, programming, testing, and maintenance. Avison & Fitzgerald (2006) also present a similar list of activities: feasibility study, system investigation, system analysis, system design, implementation and review, and maintenance.

A summary and comparison of these three viewpoints is presented in table 2. In that table, the activities have been grouped into three categories: requirement engineering, design and implementation, and validation and maintenance. These categories are presented in more detail in this section.

The activities are carried out differently in different software development organisations. When executing the activities sequentially it is called a waterfall model, and when the activities are interleaved it is called evolutionary development (Sommerville, 2007).

Table 2: Software development activities.

<table>
<tr><th>General category of activities</th><th>Haikala & Märijärvi, 2002</th><th>Avison & Fitzgerald, 2006</th><th>Sommerville, 2007</th></tr>
<tr><td rowspan="3">Requirements engineering</td><td>Feasibility study</td><td>Feasibility study</td><td rowspan="3">Software specification (Feasibility study, requirements elicitation & analysis, specification and validation)</td></tr>
<tr><td rowspan="2">Requirements analysis</td><td>System investigation</td></tr>
<tr><td>System analysis</td></tr>
<tr><td rowspan="2">Software design and implementation</td><td>Design</td><td>System design</td><td rowspan="2">Software design and implementation (Architectural design, Abstract specification, Interface design, Component design, Data structure design, and Algorithm design)</td></tr>
<tr><td>Programming</td><td>Implementation</td></tr>
<tr><td rowspan="2">Validation and maintenance</td><td>Testing</td><td rowspan="2">Review & maintenance</td><td>Software validation (Component or unit testing, System testing and Acceptance testing)</td></tr>
<tr><td>Maintenance</td><td>Software evolution /maintenance</td></tr>
</table>

Requirements engineering

Requirements engineering is a highly critical activity in the software development process. The basis for software quality, architecture, and functionality is created during it (Minkkinen & Eerola, 2007). If requirement analysis is carried out poorly it will lead to problems in later development phases.

During the **feasibility study,** it should be assessed if the software fulfils user needs, if the software is cost-effective from the business point of view, and if it can be developed within budget. The decision on whether to proceed to constructing the software or not is made by the manager on the basis of the analysts' recommendations (Avison & Fitzgerald, 2006; Sommerville, 2007).

Sommerville has named activities such as observing the existing system, discussions with users, and creating prototypes or system models to help the analysis **requirements elicitation and analysis.** Documenting the information gathered during the analysis Sommerville calls **requirements specification**. Both user requirements and system requirements are presented in a requirement specification document. Requirements are presented on a higher level to customers and end-users and on a more detailed level to system developers (Sommerville, 2007).

Requirements validation is conducted to find errors in the requirements documentation and correct them (Sommerville, 2007).

Design and implementation

The design process contains different types of **design activities** such as abstract specifications and architectural, interface, component, data structure, and algorithm design. These activities are interwoven and the feedback from one activity necessitates changes to the others. **Implementation** is the process in which requirements are converted to executable software system (Sommerville, 2007).

Design will be discussed further from the usability viewpoint in section 2.3.

Validation and maintenance

In **software validation and verification**, tests are run to verify that the software meets customer needs and follows the requirement specifications. The testing process includes component/unit testing, system testing, and acceptance testing. Acceptance testing is done with both the customer and users, and it continues until both parties agree that the software is an acceptable implementation of the requirements. The **maintenance** phase starts after validation, but because few software systems are completely new, it may be regarded that development and maintenance are a continuum (Sommerville, 2007).

The software process models presented by Sommerville, Avison and Fitzgerald, and the ISO/IEC 12207 standard do not seem to present user-centred design as a core part of the process (Figure 2), yet Sommerville states that "*careful user interface design is an essential part of the overall software design process*". Nebe et al. (2008) have also come to the conclusion that "*only little integration between usability engineering and software engineering exists*". Attempts towards integrating agile and user centred software development models will be reviewed in section 2.3.6.

2.2.2 Software process improvement

Software processes will not always lead to successful results, and thus a need arises for a new way to develop software. Seeking new ways to develop software is also necessary for software companies to survive; companies seek means to lower costs, produce higher quality products, and reduce cycle time (García-Mireles et al., 2013). Software process improvement (SPI) approaches offer ways to improve software engineering practices and aim to change existing ones (Mathiassen, Nielsen, & Pries-heje, 2001; Staples & Niazi, 2008). According to Aaen, Arent, Mathiassen, and Ngwenyama (2001), *"SPI has become one of the dominant approaches to improve quality and productivity in software engineering".*

SPI has been defined as a "set of activities that will lead to a better software process", and naturally, the goal is to deliver better quality software in time (CSIAC 2013). SPI is a complicated, systematic, and highly professional activity that requires motivated top management commitment, as well as theory, models, and skilled personnel (Wang & King, 2002). Different kinds of models for software process improvement have been presented, such as CMM (Capability Maturity Model), SPICE (Software Process Improvement and Capability dEtermination), Bootstrap, QIP (Quality Improvement Paradigm), and QSM (Quality Management System) (Aaen et al., 2001; Abrahamsson P, 2002).

Staples and Niazi have conducted a systematic review of the organisational motivations for adopting CMM-based SPI, and according to them, the main reason for organisations to adopt CMM-based SPI was to improve product quality and project performance (Staples & Niazi, 2008).

For the purposes of this study, it is important to understand what kind of methods and means are used for software process improvement in general before suggesting improvements to the HITS development process.

2.2.3 Agile software development

It has been argued that the development models in which software development activities are conducted sequentially do not comply with the needs of today's software development. According to Singh and Soni (2011), *"The old concept of sequential phase must be updated with the iteration".*

The newest and most promising models in the software industry are the agile models. Agile software development has been argued to be the winner of the future (Singh & Soni, 2011). Jokela and Abrahamsson stated in 2004 that agile software development methods and principles were "*a focus of growing interest in the field of software engineering*". Agile software development methods try to present "*an answer to the eager business community asking for lighter weight along with faster and nimbler software development processes*" (Abrahamsson et al 2002).

How can one define the agile method? Abrahamsson et al (2002) have defined the core attributes of the agile method to be speed and simplicity: develop only the primarily needed functions at first, deliver fast, collect feedback, and react to feedback. Accordingly, the agile software development method is (Abrahamsson et al., 2002):

incremental = small software releases, with rapid cycles
cooperative = customer and developers working constantly together with close communication
straightforward = the method itself is easy to learn and to modify, well documented, and
adaptive = able to make last-moment changes.

Agile thinking emphasises that requirements are difficult for users to define and users do not know what they want. It also regards that all requirements are difficult to define in the beginning of the process, so an evolutionary approach and prototyping is proposed when developing software (Avison & Fitzgerald, 2006). Requirements are then collected piece by piece, in an iterative manner.

In 2001, a group of independent thinkers of software development, "The Agile Alliance" as they call themselves, agreed on a Manifesto for Agile Software Development in which the nature of the agile model is defined. The Manifesto states as follows:

> *"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*
>
> - *Individuals and interactions over processes and tools*
> - *Working software over comprehensive documentation*
> - *Customer collaboration over contract negotiation*

- *Responding to change over following a plan*

That is, while there is value in the items on the right, we value the items on the left more." (Beck et al., 2001)

There are 12 principles behind the Agile Manifesto, presented in full in Appendix A (Beck et al., 2001).

Probably the most commonly used agile approaches are Extreme Programming and Scrum (McInerney & Maurer, 2005). According to Abrahamsson (2002), Extreme programming is more focused on practices and Scrum on the software projects. They also present other approaches: for example, Feature Driven Development (FDD) is suitable for the requirement specification phase and the Dynamic Systems Development Method (DSDN) and the Rational Unified Process (RUP) provide full coverage over the development life cycle.

In this thesis the Scrum method is reviewed because it seems to fit well to HITS development - according to Pressman (2009), Scrum is "*Good for projects that have tight timelines, changing requirements, and business criticality*". Healthcare business is critical and the requirements for both the information systems and the medical practice change continually. The Scrum method was originally presented by Schwaber and Beedle in the 1990s. According to Schwaber (1997), "*The SCRUM approach is used at leading edge software companies with significant success*".

Independent, small, and self-organizing development teams and 30-day release cycles (sprints) are the key points of Scrum according to Abrahamsson et al. (2002). The following qualities are characteristic to the Scrum method: development work is divided into packets, continuous testing and documenting is done while constructing, work occurs in sprints, requirements for the work are derived from a backlog, meetings are short, and demos are delivered to the customers within the allocated time-box (Pressman, 2009).

Scrum has been developed for managing the system development process and does not present detailed techniques for the programming phase. Management activities to identify deficiencies and impediments in the development are part of the Scrum methods, therefore Scrum helps to improve current engineering practices. There are three phases in the Scrum process: pre-game, development, and post-game (Abrahamsson et al., 2002). The phases are depicted in Figure 3.

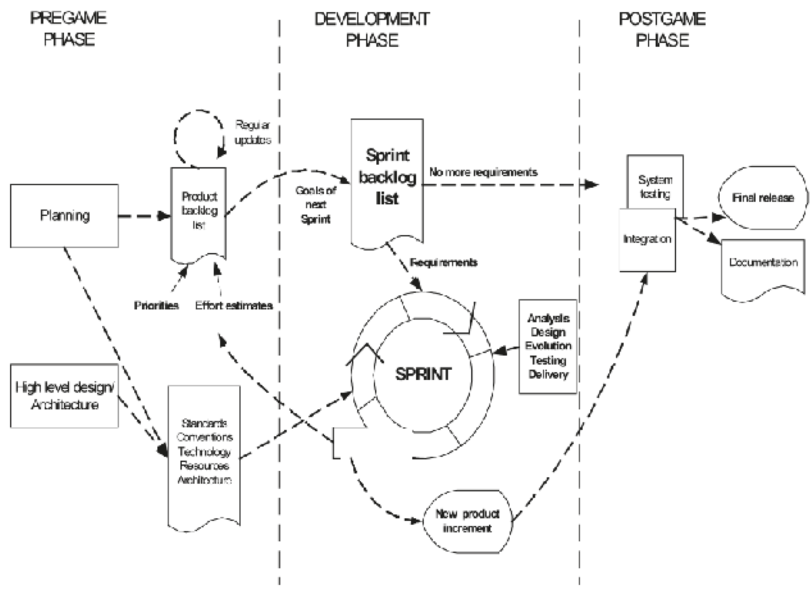


Figure 3. The Scrum process (Abrahamsson et al., 2002)

Pregame phase

As presented in Figure 3, the pregame phase contains sub-phases called ”Planning” and “Architecture/High level design”. In the **planning phase** the software is defined. If the case is about developing new software, this phase consists of conceptualisation and analysis; if the case is about a new release for an existing software system, only limited analysis and defining is done based on the current backlog. All currently known requirements are documented to the backlog (product backlog list). The requirements are prioritised and are from various sources: user, customer, sales and marketing division, customer support, or software developers. In the **architecture phase**, the current product backlog is the starting point for the high-level and architecture design of the software. In that phase it is also determined how the backlog items will be implemented. If it is not a case of new software development, possible problems that could be caused by implementing the backlog items are identified (Abrahamsson et al., 2002; Schwaber, 1997).

Development / game phase

In the development phase, the software is developed. Development activities are conducted in a pre-defined time period, which is called a Sprint. A Sprint lasts usually for one to four weeks, and there may be three to eight Sprints to complete the development. Each Sprint includes the requirements, analysis, design, evolution, and delivery phases. During the Sprint, the architecture and the design of the system evolve. A review is done at the end of each Sprint and the whole team and the product manager participate. From the review, new items could be added to the backlog (Abrahamsson et al., 2002; Schwaber, 1997).

Postgame phase

In this phase the software is ready for release. Now the requirements have been completed and the preparation for integration, system testing, and documentation are being done (Abrahamsson et al., 2002).

The knowledge about the agile development model is needed in this study in Chapter 5 when considering how the current method of developing high-usability HITS could be improved.

2.2.4 Challenges to software development

> *"Software development—in particular the development of large-scale systems that has proven so chaotic—is notable for simultaneously high uncertainty, high interdependence, and high complexity" (Adler, 2005).*

The success rate of IT projects is not very high; it has been argued that only 34% of IT projects are successful (Subramanyam, Weisstein, & Krishnan, 2010).

According to Fægri (2011), creativity and problem solving skills are determining factors when building software. Understanding the user needs, use, and the use context is also extremely important. The quality of communication in the project was also seen as essential. It has been argued that end user involvement is one of the most important success factors in end user satisfaction and quality in IT projects (Høstgaard et al. 2011; Lilja, 2013; Pekkola et al. 2006; Subramanyam et al., 2010). In chapter 2.4, the involvement and participation of users in the software development are examined in greater detail.

Software development differs depending on whether the case is project-based or product-based development. According to Dittrich (2014), the development and evolution of software products is not well supported by the software engineering tools, techniques, or methods. She also states that the discipline of software engineering takes projects as the main organisational form of software development. The problem, then, is in how to apply mainly project-oriented methods to product-oriented software engineering.

In the world of software design and development, the concept of generification has been raised lately. Generification is the case in which a locally developed software product is marketed toward larger audiences. According to Johannessen and Ellingsen (2009), typical challenges and questions used to judge when generification of a software product is needed are:

- What are particular requirements from a few customers and what are general ones?
- How particular are those requirements from the few customers? Should diversity be built into the system or should functions meeting these particular needs be customised at each site?

Thus, generification also brings challenges to software product development.

Yet another challenge for software development is 'software ecosystems'. The software ecosystem approach is an emerging trend (Bosch, 2009). Hanssen (2012) describes the software ecosystem as follows: *"This new concept and its implicit reference to ecology imply a shift of focus from the internals of the software organization (the individual organism) towards its environment and the relations and actions within (the ecosystem)"*. The key challenges that software ecosystems pose for software engineering, according to Dittrich's research, are: keeping contact with other actors in the ecosystem, establishing techniques to support multilevel development and evolution, managing an overlay of development cycles with different rhythms, and maintaining documentation and modelling support for continuous development. (Dittrich, 2014)

According to Bosch and Bosch-Sijtsema (2010), we can see a clear path from software product lines to software ecosystems – an organisation has a platform, and internal developers build products on top of that. The software ecosystem enlarges this; external developers or developers' communities build on top of that same platform, and domain experts and users are included in development, which requires new means for collaborating and coordinating development (Bosch & Bosch-Sijtsema, 2010).

2.2.5 Broaden the perspective: The ADISD

This study was conducted within a research group in Information Systems as a social science of information and technology, DAISY (DAISY, 2015). This study does not focus on the use of information and IT in user organisations, but nevertheless shares the basic premises of the group. Therefore this section will not discuss Information Systems issues in general, but rather just present the inputs from this background that will be used later in the study. This section complements the theoretical discussions in Papers III and IV.

An information system is defined by Pentikäinen (2014) as *"a collection of all the elements which offer the right information at the right time in the right place so that available information meets the needs of"* . And also by (Luukkonen et al., 2013, p. 449-450). *"the processes of managing (creating, using, storing, exchanging, etc.) information in an organizational setting (in work activities) for a purpose" – "a socio-technical entity in the user organization consisting of people (actors), information (contents), and technology (means), linked together by a process directed toward a purpose"* To develop an information system (information system development, ISD) is, in that sense, a wider and more complex process than that used to develop just a software product (software engineering, software development).

The focus of this thesis is to present the Activity-Driven Information System Development (ADISD) model of the DAISY research group to broaden the perspective of software development and have a better understanding of the simultaneous development of work activities and software.

ADISD is a three-level analysis model for shared understanding in ICT development work. The model was developed and tested in the context of healthcare (e.g. in Finnish healthcare projects and organisations). The objective for developing the model was to narrow the distance between work improvement and information system development (Toivanen et al., 2007; Toivanen, Luukkonen, & Mykkänen, 2009; Pentikäinen, 2014). The Scandinavian ISD approach, user participation, was emphasised in developing the model.

The theoretical background of the model is in Activity Theory and more specifically in the Activity Analysis and Development (ActAD) framework. According to Korpela et al. (2000, p. 207), Activity Analysis and Development "… *is a comprehensive framework and methodology that guides developmental efforts to be broad-based, rather than limited to information technology only*".

The ADISD model facilitates the perception of a big picture view of the subject of development in context and the effects of the planned changes (Toivanen et al., 2007; 2009; Pentikäinen, 2014). The developers of the model see 'the information system' as only one element of a work activity, which consists of both manual and automated parts and the users.

The ADISD model consists of three levels and three phases (see Figure 4). In order to achieve an information system which fulfils the needs arising from a work activity, the present state analysis is needed to recognise the needs for development. The analysis for a shared understanding of the present state is the first phase of the ADISD model. A shared understanding of the goal state is the purpose of the second phase. Everyone should understand how the outcome of the development will affect their work. The third and final phase is called 'making the plans of change', meaning a practical plan for how the jointly defined goal state will be achieved. It is important to understand that the phases of the model are to proceed iteratively, moving back to earlier phase if there is a need to reconsider something (Toivanen et al., 2007; 2009; Pentikäinen, 2014).

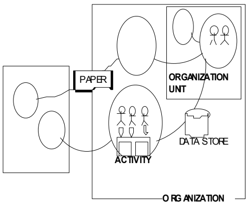
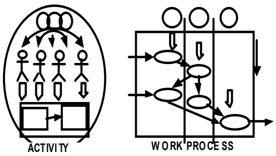
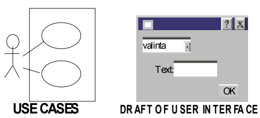
Phases / Levels & Describing models	1. **Analysis** for shared understanding of **the present state**	2. **Design** for shared understanding of **the goal state**		3. Making **the plans for change**
1. Network of activities & Information landscape 	Overview, what services or products do we produce and for whom, who are our stakeholders, what are the essential activities? What are the most important information entities in our activities, where are these information entities located, how do we communicate in the network, and which tools are used?	In information system development (e.g. the introduction of new or tailored software or an integration project), we have to outline how and where change affects the network of activities and information landscape. Zoom out from processes and information flows to see the range of changes.		Considering - context (buildings, infrastructure, legislation etc.) - changes in the network of activities and - changes in the information landscape. Planning - re-organized services and activities - purchase of software and hardware and - systems integration.
2. Work activity (processes) & Information system (data flows) 	Zoom in on our essential activities: who is involved in the work process, who makes decisions in different stages, how is the work coordinated, what means (mental and physical) are needed in the process, what information is needed and where from, what is written down, and what information tools are used?	In ISD (e.g. the introduction of new or tailored software or an integration project), we have to outline how and where we utilize IS in work processes, what our work processes will be after the change, and what impact change has on the information systems and data flows. Zoom out from the actions (and use cases) and information tools to see the range of changes.	**Validation, verification, decisions**	Planning - changes in the information system - changes in work processes e.g. introduction of new software and new work practices step by step, unit by unit
3. Actions & Information tools 	Zoom in on our essential work processes, what actions need to be developed, and what detailed information sets and data items are needed in central actions. What information tools (forms etc.) are needed? We can outline use cases from the actions if software is used.	In information system development, we have to outline how we utilize the IS that will be developed in actions, how it would be used, and what effects the changes have on information tools. Users' needs, wishes, and requirements must be considered. We can specify use cases from the actions if software is or is-to-be used.		Planning - changes in information tools (e.g. software) and their use, - changes in actions and duties.

Figure 4. The ADISD model in a table form (Toivanen et al., 2009)

The levels of the model are: 1. activity network and information landscape, 2. work activity and information system, and 3. actions and information tools. The information system is studied in the work context on each level. On the first, highest level, the goal is to capture an overview of the domain. On the second level, workflows and information as well as information tools are mapped, and the lifecycles of information entities are identified. A holistic picture of the activities should be obtained. The third level is for the most detailed analysis. Individuals' actions and the tools they use are the focus of analysis on this level. User interface drafts and prototypes are used to gather detailed information. When applying the model, each upper level can be used as a map that the lower level can be reflected on

– it is possible to zoom in and out in the descriptions (Toivanen et al., 2009; Pentikäinen, 2014).

The experiences of the developers show that the model makes it easier to co-develop changes in work and in information systems (Luukkonen et al., 2013; Pentikäinen, 2014). According to Toivanen et al. (2009), the best times to use this model are in the early phases of the ISD process (domain analysis & requirement elicitation), when introducing new software in an organisation, and when new work practices and processes are planned with the support of new software.

2.3 USABILITY

"Focus on What Users Need, Not What They Want." (Eastwood, 2013)

This section deals with the primary background theory of the thesis. The basic concepts of usability, user interface, and user experience are first investigated in sub-section 2.3.1. The next three sub-sections review literature on usability design; first on a general level (sub-section 2.3.2), then zooming in to the closely related traditions of User-Centred Design, Interaction Design, and Participatory Design (sub-section 2.3.3). Lastly, two practical user-centred design models for software development are presented (sub-section 2.3.4). Service design is a more recent design approach which is reviewed in sub-section 2.3.5. Finally, sub-section 2.3.6 reviews literature on integrating UCD with the agile model.

The literature review in this section complements the reviews included in all Papers I-IV. It is assumed that the reader has already familiarised herself or himself with the Papers.

2.3.1 Basic concepts: Usability, user interface, user experience

Usability

The ISO 9241-11 standard definition of usability was already quoted in section 1.2.2: *"extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"* (ISO, 2009 p. 3).

As the standard definition stresses, the context in which a software system is used is significant when assessing its usability. In addition, Bevan emphasises the viewpoint of specific user groups: *"…if different groups of users have different needs, then they may require different characteristics for a product to have quality for their purposes"* (Bevan, 1997). Usability is a quality attribute, but as Bevan emphasises: *"Products can only have quality in relation to their intended purpose"* (Bevan, 1997).

Nielsen defines usability by five quality components: learnability, efficiency, memorability, errors, and satisfaction. In addition to usability, there are many other equally important quality attributes of software. One of these is the utility, which for Nielsen means the design's functionality from the user's viewpoint. In Nielsen's terminology, usability and utility together determine whether a software system is useful (Figure 5) (Nielsen, 2012).

Utility	whether it provides the **features you need**
Usability	how **easy & pleasant** these features are to use
Useful	**usability + utility**

Figure 5: Usability, utility, and usefulness according to Nielsen (2012).

In this study, usability is seen more widely than in Nielsen's definition, as already mentioned in section 1.2.2. This study agrees that, fundamentally, usability is about supporting users' work (Jokela, 2015).

User interface

User interface is the only part of a software system that a user is able to see, and therefore it is an extremely important part of the software. There are several definitions of user interface, two of which are presented here. Firstly is the Tech Terms Computer Dictionary definition:

> *"A user interface, also called a "UI" or simply an "interface," is the means in which a person controls a software application or hardware device."* (Christensson, 2015)

Secondly is the definition from BusinessDictionary:

> *"Visual part of computer application or operating system through which a user interacts with a computer or a software. It determines how commands are given to the computer or the program and how information is displayed on the screen."* (WebFinance, 2015)

The information displayed on the screen, the content, is important when thinking about the usability of software. Content needs to be understood and respected when designing a user interface. Halvorson quotes Jeffrey Zeldman's thoughts about the significance of content:

> *"Content informs design; design without content is decoration. Content has the same relationship to design that product has to advertising. Good ads are based on the product; good designs come from and facilitate the content. This is one reason we bring content strategy to every design assignment, and one reason we insist on working with real content, not lorem ipsum (placeholder) content. Nothing is sadder than a beautiful design that works great with lorem ipsum but doesn't actually support the real content."* (Halvorson, 2010 p.77)

Jokela (2015) likewise stresses that a visually attractive user interface does not mean good usability, and good usability is based on properly designed software structure – not on user interface design.

User experience

User experience is seen by Bevan (2008) as one more approach to usability and is defined as follows:

> *"User Experience (satisfaction): Meeting user pragmatic and hedonic goals related to the experience and outcomes of interaction."* (Bevan, 2008, p. 1)

Jokela argues that it would be obvious to define usability as objective user performance and user experience just as it is: subjective user experience (Jokela, 2015).

Hassenzahl and Tractinsky (2006) claim that user experience (UX) has become a buzzword in HCI (human-computer interaction). Interactive products have become fascinating things to be desired, in addition to being useful and usable. UX as term "*promises change and a fresh look, without being too specific about its definite meaning*" (Hassenzahl & Tractinsky, 2006, p. 91).

A common definition of user experience has been pursued in recent years, and one such definition is presented in the ISO standard 9241-210: "*A person's perceptions and responses that result from the use and/or anticipated use of a product, system or service*" (ISO, 2009, p. 3). Several other definitions also appear (see Appendix B).

Law and Roto recommend the term user experience to be used about products, systems, services, and objects that a person interacts with through user interface (Law & Roto, 2009).

Hassenzahl & Tractinsky argue interestingly that UX "*focus[es] on how to create outstanding quality experiences rather than merely preventing usability problems*". They compare the assumption that 'absence of illness equals health' to the assumption that absence of problems means good usability and user experience. In their opinion, the main future goal is to "*contribute to our quality of life by designing for pleasure rather than for absence of pain. UX is all about this idea*" (Hassenzahl & Tractinsky, 2006, p. 95).

User experience is seen as a broader concept than usability. *"'User experience' encompasses all aspects of the end user's interaction with the company, its services, and its products"* (Nielsen & Norman, 2015).

2.3.2 Designing for usability

In order to design usability and user experience for interactive products, a designerly way of thinking is necessary. Design thinking has been defined as follows:

> *"… a methodology that imbues the full spectrum of innovation activities with a human-centered design ethos. By this I mean that innovation is powered by a thorough understanding, through direct observation, of what people want and need in their lives and what they like or dislike about the way particular products are made, packaged, marketed, sold, and supported."* (Brown, 2008, p. 1)

According to Norman (2013), human-centred design is a powerful tool of design thinking.

To design good usability, user-centred design methods are widely used and assumed to be the key for studying user needs and building products with good usability. User-centred design (UCD) methods emphasise user participation in the design and development process. UCD is a multidisciplinary design approach, in which users are actively involved in order to understand the user and task requirements. UCD methods are iterative in nature; i.e., design and evaluation are repeated until a desired goal is achieved (Mao, Vredenburg, Smith, & Carey, 2005). According to Rogers, Sharp, and Preece (2011), three basic, commonly accepted principles of the user-centred approach (originating from Gould and Lewis, 1985) are:

1. Early focus on users and tasks
2. Empirical measurement
3. Iterative design

There are a number of methodologies and approaches for user-centred design (e.g., usability engineering, human-centred design, interaction design, and participatory design), each one being slightly different from the others.

Sanders (2006) has presented a cognitive collage of the design research space as in 2006, which helps to put things into perspective (Figure 6).

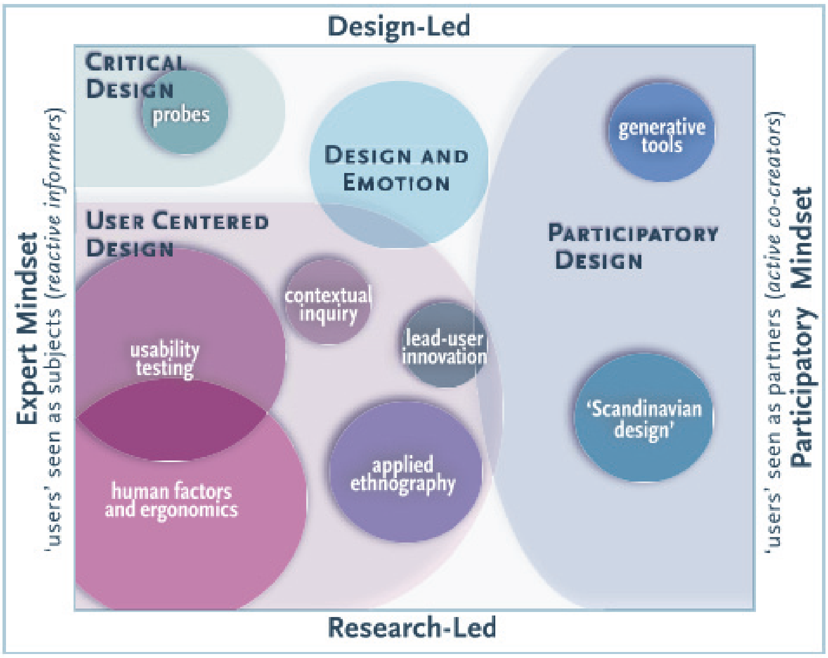


Figure 6. Sanders's topography of design research (Sanders, 2006, p. 4).

In the figure, the top half of the graph (design-led) contains design research methods and tools that have been introduced from a design practice background, and the bottom half (research-led) contains those from a research background. The figure is horizontally divided according to the mindsets of those who practice and teach design research. Expert mindset and participatory mindset are seen as two distinct cultures of design research. In the expert mindset, users are seen as subjects or reactive informers, and in the participatory mindset they are seen as partners and active co-creators. Service design (reviewed in sub-section 2.4.5) would be positioned near the middle in order to draw upon tools and methods from all parts of the collage (Sanders, 2006).

Keinonen (2009) has also presented a framework called Design Contribution Square (DCS) for organizing human-centred design approaches. The framework is for mapping human centred design practices and methods based on designers' and users' activity (Figure 7). The meanings of the abbreviations in the figure are as follows:

U_{pro} proactive user contribution
U_{re} reactive user contribution
U_{in} inactive user contributions
D_{pro} proactive designer contribution
D_{re} reactive designer contribution
D_{in} inactive designer contribution

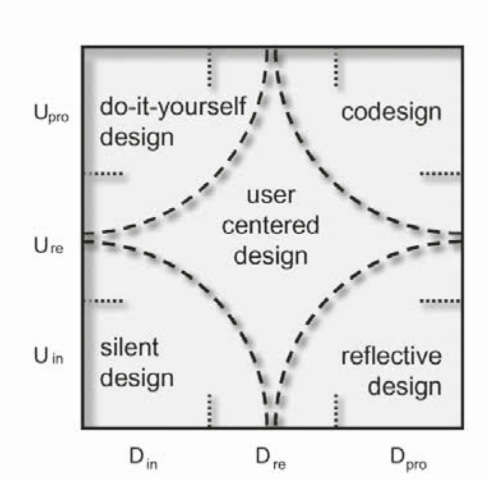


Figure 7: Design Contribution Square (Keinonen, 2009)

Sanders, like many other researchers, regards UCD as an expert-driven approach in contrast to Participatory Design, while Keinonen regards UCD as the core of different human-centred design approaches. In this thesis UCD is used in the latter way, as an umbrella term for different methods and approaches in which the user's "presence" in the development process has the central role. Abras, Maloney-Krichmar, and Preece (2004) state:

> *"'User-centered design' (UCD) is a broad term to describe design processes in which end-users influence how a design takes shape. It is both a broad philosophy and variety of methods. There is a spectrum of ways in which users are involved in UCD but the important concept is that users are involved one way or another."* (Abras, Maloney-Krichmar, & Preece, 2004, p. 1)

Human / user aspects of development are seen as essential from the end-users' perspective and for system success (Maguire, 2001; Rogers, Sharp, & Preece, 2011; Shah & Robinson, 2006). In addition to the positive effects of applying user-centred design in software development in general, in the medical domain the UCD process is required to be adopted by the ISO/IEC standard 62366: *Medical Devices – Application of Usability Engineering to Medical Devices* (Hodgson, 2010).

2.3.3 User-Centred Design, Interaction Design, and Participatory Design

Three major traditions or "brands" of designing for usability are given a closer look in this sub-section.

User-Centred Design

In this thesis, the term user-centred design is used in the way defined by Mao et al. (2005):

> *"User-Centered Design (UCD) is a multidisciplinary design approach based on the active involvement of users to improve the understanding of user and task requirements, and the iteration of design and evaluation."* (Mao et al., 2005, p. 105)

The ISO standard 9241-210:2010 *Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems* that replaced the well-known earlier standard ISO 13407 describes human (user) centred design (Jokela, 2010). The standard provides "*requirements and recommendations for human-centred design principles and activities throughout the life cycle of computer-based interactive systems*" (ISO, 2009). The standard is intended to be used by people who manage design processes (ISO, 2009). It defines the following four core activities: understanding and specifying the **context of use**, specifying the **user requirements**, producing **design solutions** to meet user requirements, and **evaluating** a design against the requirements (see Figure 8).

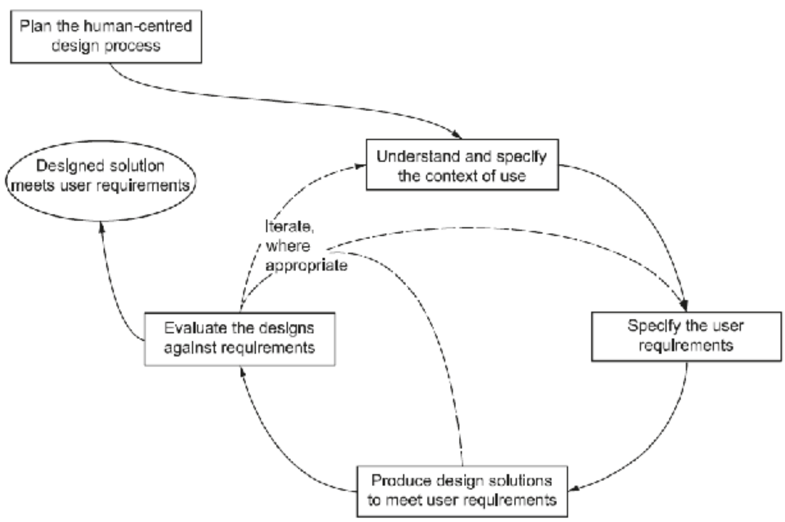


Figure 8: Human-centred design process according to ISO 9241-210 (ISO, 2009, p. 11)

According to Bevan (1997), *"The purpose of designing an interactive system is to meet the needs of users: to provide quality in use"*. To meet the needs of users, it is crucial to understand and specify the context of use and to specify the user requirements, the first two steps in the iterative human-centred design presented in the ISO standard.

When producing design solutions to meet those user requirements, the secret to success according to Norman (2013) is to understand what the real problem is. Evaluating the design against the requirements in the evaluation phase is extremely important. According to Jacob Nielsen (2001), it is not evaluation to simply show designs to users and ask which one they prefer; users have to try to use a design to be able to form an opinion.

Interaction Design

Very similar activities are presented in the interaction design process model by Rogers et al. (2011). The model has four basic activities, which are supposed to be repeated: establishing requirements, designing alternatives, prototyping, and evaluating (Figure 9). Interaction design is defined by Rogers et al. as "*designing interactive products to support the way people communicate and interact in their everyday and working lives*" (Rogers et al., 2011).

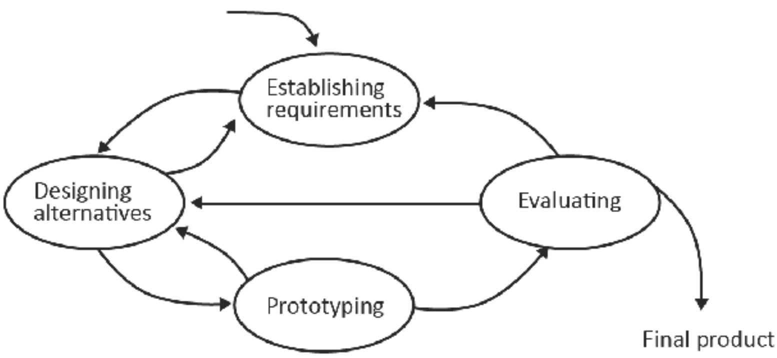


Figure 9: Interaction design lifecycle model (Rogers et al., 2011)

Participatory Design

Participatory Design (PD) is an originally Scandinavian approach to designing information systems with users, an approach to design in a user-centred way. Muller (2003), drawing from Greenbaum & Kyng (1991) and Muller & Kuhn (1993), defines Participatory Design as *"a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer based activities"* (Muller, 2003 p. 1062). According to Hansson, Dittrich, and Randall (2006), the Participatory Design approach has two important characteristics: a political nature (questions about workplace democracy and control over the workplace), and a technical nature (the participation of skilful users in a development process contributes to successful design and high quality products).

There are a number of guiding principles and techniques in the Participatory Design approach. Hansson et al. (2006) list the principles and methods as follows:

Principles of participatory design:

- the acknowledgement of diversity
- ensuring the real impact of the participants' viewpoints in the design process
- mutual learning at all levels
- separation of interest groups
- protection of participants

Methods of participatory design:
- visiting work places
- creating work scenarios
- future workshops and organisational games
- mockups
- prototyping

2.3.4 Practical user centred design models and techniques

In this sub-section, the focus moves from general approaches to practical methods and models. A collection of UCD techniques and methods is presented first. Two practical UCD models for software design are then presented briefly. The first one is the well-known Goal Directed Design model and the other is a practical goal-based design model called the GUIDe process model. These models were selected because they both stress users' goals as the starting point of design, which presumably fits healthcare IT system development well.

UCD techniques and methods

A large scale mapping of UCD techniques and methods to the core UCD activities presented in the ISO standard 13407 (which in 2009-2010 was replaced by ISO 9241-210) has been conducted by Maguire (2001) (Table 3).

Table 3 : A set of methods or activities that support the core activities presented in the ISO standard 13407, adapted from Maguire (2001).

Activities	Methods
Planning	Usability planning and scoping Usability cost-benefit analysis
Context of use	Identify stakeholders Context of use analysis Survey of existing users Field study / User observation Diary keeping Task analysis
Requirements	Stakeholder analysis User cost-benefit analysis User requirements interview Focus group Scenarios of use Personas Existing system/competitor analysis Task/function mapping Allocating of function User, usability and organisational requirements
Design	Brainstorming

	Parallel design Design guidelines and standards Storyboarding Affinity diagram Card sorting Prototyping (Paper, Software, Wizard-of-OZ, Organisational)
Evaluation	Participatory evaluation Assisted evaluation Heuristic or expert evaluation Controlled user testing Satisfaction questionnaires Assessing cognitive workload Critical incidents Post-experience interviews

According to Mao et al. (2005), UCD methods have been criticised to be ineffective, impractical, too complex, too time-consuming, and too expensive to use in practice. Likewise, Humayoun et al. (2009) state that software teams hesitate to use UCD activities because they are considered time-consuming and effort-intensive. However, nowadays "*the growing popularity of e-commerce has greatly bolstered the appeal of usability and UCD, as users can take their business elsewhere with just one mouse click*" (Mao et al., 2005, p. 106).

Goal-Directed Design

According to Cooper, Reimann, and Cronin (2007), user research has been noted as an important part of the development process, but most development methods do not provide means to translate user research findings into design solutions. The aim of the Goal-Directed Design (GDD) model is to narrow that gap.

The GDD process has six phases: research, modelling, requirements, framework, refinement, and support (Figure 10). The phases are briefly described in the following (Cooper et al., 2007).

The process starts with **research** on the users and the domain. Ethnographic field study methods like observation and contextual interviews are utilised. The scope of the project is also defined, competitive product audit and market research are conducted, and stakeholders like developers, subject matter experts, and technology experts are interviewed. One of the outcomes of this phase is a set of behavioural patterns, which helps to categorise the modes of use of the software being designed. Personas, created in the next phase, need these behaviour patterns as a starting point.

In the **modelling** phase, users and the context are modelled**.** Domain models (e.g., information flow and workflow diagrams) and user models (i.e. personas) are created on the basis of the research. Understanding the importance of tasks and the reasons why they are important is achieved by creating personas.

Requirements definition of the user and business and technical needs is then conducted. Personas are the main characters of context scenarios created in this phase. **Framework definition**, i.e. a definition of the design structure and flow, is created by the designers. An overall product concept is created, defining the basic framework of the software behaviour and the visual design. The output of this phase is an interaction framework definition.

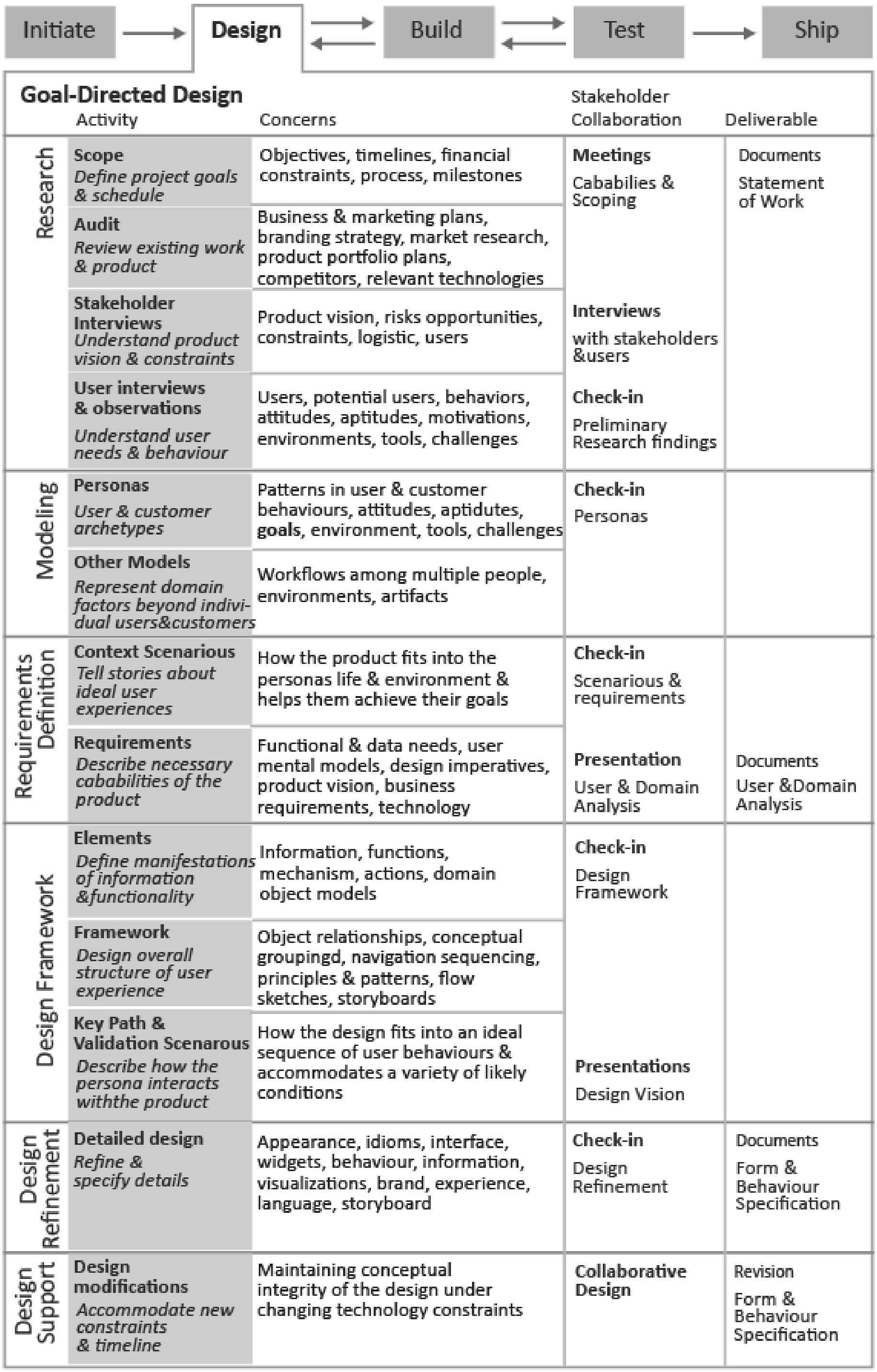


Figure 10. Goal-Directed Design phases (Alan Cooper et al., 2007).

In the **refinement phase** the focus is on details and implementation. Designers conduct walkthroughs and validate scenarios to ensure task coherence. In the last phase, **development support,** interaction designers' participation is important to prevent developers from compromising the design when implementing the software.

GUIDe

Another goal-based model is the GUIDe process model (Goals - User Interface Design – Implementation), in which requirements are visualised into a series of screenshots. In the GUIDe process model user interface (UI) design is input for and sets requirement to the system architecture or database design. Thus, the model emphasises the need for user interface design in the beginning of the development process (Laakso & Laakso, 2004).

The starting point of developing this model was the assumption that requirements can be established exactly enough with the help of appropriate methods in the beginning of the development process. The GUIDe model phases are: definition of goal-based use cases based on realistic use situations; simulation-based design and testing of the user interface; and producing user interface specifications to the implementation phase. User interface specifications visualise the requirements specification in a more unambiguous way to all stakeholders, and naturally set the requirements for designing the implementation (Laakso & Laakso, 2004; 2015).

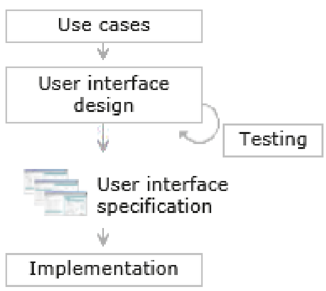


Figure 11: GUIDe phases (Laakso & Laakso, 2015).

Use cases are based on real life use situations, which are found out by field studies: user observations and contextual interviews. These use situations are then analysed and formed into goal-based use cases.

Field studies produce a lot of use situations, but in the Laaksos' experience these seem to be reducible to about 3-6 categories. When writing goal-based use cases, the designer needs to see a concrete goal above the system. User interface designs are created on the basis of the use cases by simulating the use cases. The designer draws UI sketches and simultaneously designs the interaction between the user and the software. Low-fidelity paper prototypes are drafted.

Designs and paper prototypes are tested with users by using usability test methods or the lighter and more flexible usability walkthrough method. It is easy to make changes since no code has been written yet. After the design and testing, user interface specifications are written. The specification describes how the software looks and how it behaves. The GUIDe model does not deal with the technical implementation.

2.3.5 Service design

Miettinen et al. (2014, p. 45) summarise service design as *"a new competence area that helps in managing and developing service experiences in various contexts and themes of interest*". Accordingly, service design is an interesting latest addition to the user-centred design approach.

An iterative process familiar from UCD, including prototyping, testing, analysing and refining work, applies well to the service design process. The aim of service design is to ensure that service interfaces are useful, usable, and desirable from the customers' viewpoint, effective and efficient from the service provider's viewpoint, and distinctive from the suppliers' viewpoint. Service design is characterised by the creation of value and by an interactive process (Mager, 2009; S. Miettinen, 2009).

Holmlid has studied service design methods and user-centred design practices, and concludes that "*Service Design provides an overall design and contextualizes Interaction Design for technology supported services. The focus on users and the method portfolio are shared, which provide user-centred design action space across not only a technology development process, but also an organizational change process.*" (Holmlid, 2005)

According to Miettinen (2009), the service design process has variation, but it has started to find its form. Five factors need to be known when utilising service design (S. Miettinen, 2009):

- Understanding the service design challenge; the users, business environment, and applicable technologies.
- Observing, profiling, creating empathy for the users, participating with the users, and being visual during the whole process.
- Creating ideas, prototyping, evaluating, and improving the act of including the clients and the users in the process.
- Implementing, maintaining, and developing services.
- Operating with business realities.

Service design is a holistic approach to designing services, and often necessitates interdisciplinary work. Software engineering among other research fields such as design, experience design for services, and service marketing, can utilise the service design methods and tools (S. Miettinen et al., 2014).

2.3.6 Agile user-centred design

According to Da Silva et al. (2015), the term Agile User-Centred Design means that user-centred design is used in agile environments. User-centred design and agile methods emphasise the importance of users being a major part of the development activities. One of the agile development principles is cooperative development, which means that the customer and developers are working continuously together with close communication (Abrahamsson et al., 2002). The third principle of the Agile Manifesto is "*Customer collaboration over contract negotiation*" (Appendix A).

The successful integration of agile methods and UCD most likely brings about benefits to users and businesses (Chamberlain, Sharp, & Maiden, 2006). Many researchers have stated that agile and user-centred design are compatible (e.g., Chamberlain et al., 2006; McInerney & Maurer, 2005). Agile methodologies themselves do not provide instructions on how to integrate UCD and software engineering work (Kuusinen, Mikkonen, & Pakarinen, 2012). However, a few models of integrating agile and UCD methods have been presented (Da Silva et al., 2015; B. D. Fox, 2010; D. Fox, Sillito, & Maurer, 2008; Humayoun et al., 2009).

A systematic review of UCD and agile methods by Da Silva, Martin, Maurer, & Silveira (2011) shows that the two are a natural fit, but more controlled experiments on integrating them are needed. Maintaining a Big Picture was regarded as very important, while iterative development was noticed to be difficult. Da Silva et al. recommend that when UCD and agile methods are integrated, the focus should be on both design and usability evaluation. The integration model resulting from the systematic review is presented in Figure 12 (Da Silva, Martin, Maurer, & Silveira, 2011).

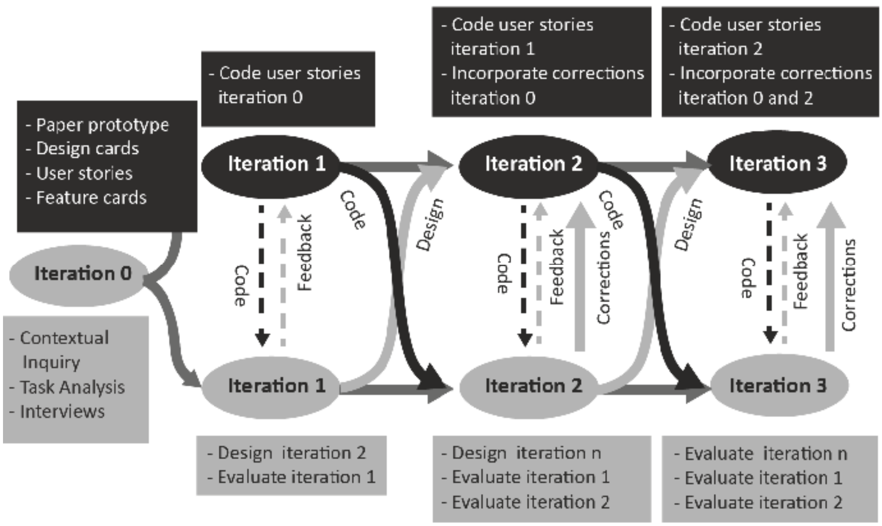


Figure 12. High-level UCD and agile integrated process, the result of the systematic review by Da Silva, Martin, Maurer, & Silveira, 2011.

The importance of conducting a big design in the beginning of the development process is an issue on which these two methodologies differ. Agile methods do not recommend up-front design, while UCD does, although both methodologies are iterative in nature. In the agile community's opinion, up-front big design is not needed because business and user requirements will change during the implementation (Beyer, 2010). On the other hand, successful projects that have significant user interfaces and a significant impact on users' work practices have found up-front design necessary (Beyer, 2010). Laakso & Laakso (2015) also stress the need of an up-front big design:

> *"We cannot design a good user interface one piece at a time, because we have to be able to perform all the main tasks with the same user interface from beginning to end. If features A and B, which are needed to perform task 1, are planned at different stages, using these features after each other may not be straightforward. The transition from one feature to another may demand a lot of navigation from the user, or the data produced by feature A is not visible at the same time as the data from feature B, though the user may need to compare the two."* (Laakso & Laakso, 2015)

When integrating UCD and agile models, an once-only initial stage or iteration 0 should be added for defining the scope and conducting UCD (e.g., user research, UI design and low-fidelity prototyping) (Hugh Beyer, 2010; Da Silva et al., 2011; D. Fox et al., 2008). The UCD process is proposed to go at least one sprint ahead of the agile process, as Figure 12 also shows.

According to Hugh Beyer (2010) it is not clearly understood by developers how much time good UI design takes, and most development processes are designed for developers and do not take into consideration the time for UCD work. This is problematic also in the agile development model.

2.4 USER PARTICIPATION

Alan Cooper et al. (2007) present three main reasons for poor design: ignorance about users; conflict of interest between serving human needs and constructing priorities; and the lack of a process for understanding human needs as an aid to developing appropriate product form and behaviour.

In user-centred design methods, users have an extremely important role in the development process of successful software. Indeed, user involvement and user participation are the key issues when measuring success in information system development projects (Keil & Carmel, 1995; Lynch & Gregor, 2004). The CHAOS Manifesto 2013 has listed 10 success factors for projects, and user involvement is listed in second place (Standishgroup, 2013). Holmlid has also argued that user participation enriches technology development, instead of hindering or slowing it down (Holmlid, 2005). Bødker & Greenbaum (1993) state that the issue is not about whether users should be involved, it is about how users' knowledge and experience can be utilised better.

Sanders has presented a history of the changes in the way designers have thought about people (Figure 13). Her idea is that, today, "*we invite the people we serve through design to participate with us in the actual designing. We are now beginning to think of people as participants in the design process, as adapters of the designed artifact or even as co- creators, i.e., equal in stature and possessing of unique and relevant expertise. At the top of the "hill", designers become interpreters of people's needs and dreams and not just the creators of artifacts.*" (Sanders, 2005)

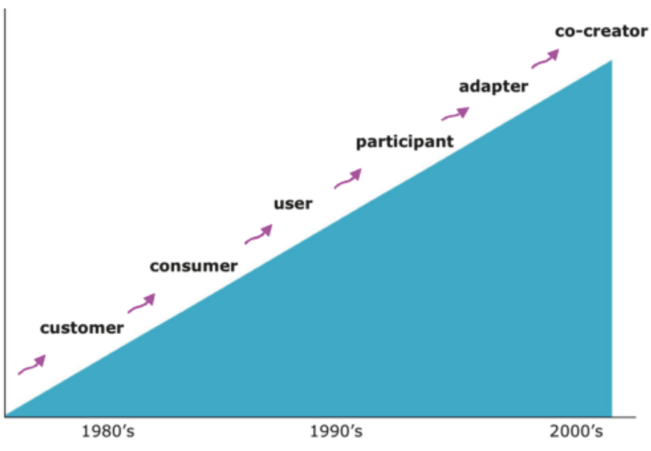


Figure 13: History of how designers think about people (Sanders, 2005)

Software usability design without users is guesswork. Mattia and Weistroffer (2010) have identified, by reviewing a number of studies, general propositions of the theory that a successful user participation approach suggests: (1) a system solution that improves quality, (2) a system solution that increases user acceptance, (3) a system solution that increases user satisfaction, (4) a system solution that improves budget and schedule performance, (5) a system solution that increases feelings of ownership, and (6) the presence of change agents (administrative roles) in the formal social network of participants.

2.4.1 User participation versus user involvement

The concepts of user participation and user involvement are not always clear: user involvement and user participation have different meanings. According to Hwang and Thorn "b*oth user involvement and user participation are beneficial, but the magnitude of these benefits much depends on how involvement and its effects are defined*" (Hwang & Thorn, 1999). Lin and Shao state that user participation is *"conceptualized as a behavioral construct (the degree of participative behaviours of users during the development process)"*, while user involvement and user attitudes are *"psychological constracts"* (Lin & Shao, 2000). Barki and Hartwick also have separate constructs of user participation and user involvement (Barki & Hartwick, 1989; 1994). They define user participation as *"a set of behaviors or activities performed by users in the system development process"* and user involvement as *"a subjective psychological state reflecting the importance and personal relevance of a system to the user"* (Barki & Hartwick, 1989). They also noticed that "*users who participate in the development process were likely to develop beliefs that a new system is good, important, and personally relevant"* (Barki & Hartwick, 1994, p. 75).

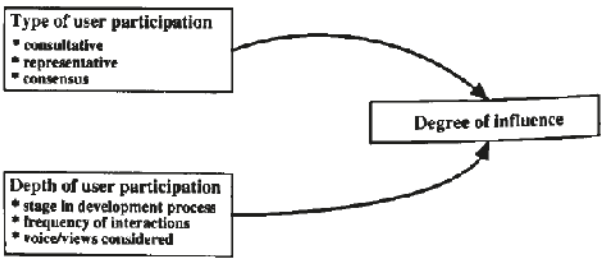


Figure 14: User influence on system features (Lynch & Gregor, 2004).

Lynch and Gregor argued that the concept 'user participation' is too simplistic, and brought forward the construct 'degree of user influence on design' in their study. According to them, this depended on two aspects of system development, which were the type of user participation and the depth of participation (Figure 14). The types of user participation are consultative, representative, and consensus. The depth of user participation contains three aspects: stage in development process, frequency of interaction, and voice/views considered (Lynch & Gregor, 2004).

2.4.2 Early participation

According to Kujala (2002), understanding customer need, particularly in the beginning of the development process, in the requirement definition phase, is important and seen as a success factor. Bostrom's study (1989) supports the claim that effective communication between developers and users is important in the early phase, particularly in the requirement definition phase. If users are involved only in the later phases of development, it may lead to a situation where user requirements cannot be taken into account without redesigning the system widely (Lynch & Gregor, 2004). According to Jokela (2011), users' needs must be understood even earlier, in the procurement process, and the responsibility of gathering user needs and requirements belongs to the supplier.

2.4.3 Software system success and user participation

Lin & Shao's (2000) study shows a positive link between system success and user participation, and they argue further that system complexity has a major effect on user participation. Harris & Weistroffer (2008) also emphasise that user participation in the development process is critical to system success.

User participation in development is not always straightforward and clear; there are challenges to user involvement. In 1984, a review of empirical research by Ives and Olson showed that the benefits of user involvement had not been strongly demonstrated – only 8 out of 22 studies showed a positive link between system success and user involvement, 7 showed mixed results, and 7 showed negative or insignificant results. They also brought up concerns about the need to develop and validate measurement and methodology (Ives & Olson, 1984).

A more recent study has also discussed that the findings of links between system success and user participation are not consistent; the authors suggest that a reason for that may be the difficulties, vagueness, and many dimensions of the concept of participation (Cavaye, 1995). Tesch et al (2009) argued that the effects of user participation may also depend on users' knowledge or skills in the information system domain. *"Close Co-operation with the Customer Does Not Equal to Good Usability"*, conclude Jokela and Abrahamsson in their 2004 study, in which both the customer and users were involved but the usability maturity level was low.

Harris & Weistroffer (2008, p. 3) have stated that *"The ascertained impact of user participation on system success seems to have been greater in the more recent studies. It is possible that user participation has become more directed and thus more effective"*. They have also presented six points to help information systems professionals choose the right kind of user participation to achieve maximum benefits (Table 4).

Table 4 : Six points on user participation (Harris & Weistroffer, 2008)

No	Point
1	**User participation has the greatest impact on system success if the user is allowed to voice an opinion and make choices from predefined options.** The reasoning may be that with the voice and choice option, users anticipate their opinions and concerns to be accepted and implemented by the developers, thus raising their confidence and satisfaction levels. This constitutes a kind of shared user participation: the users feel like partners in the development process, having a sense of control over the outcome.
2	**The importance of user participation increases with system complexity.** The explanation may be that more complex systems make the determination of system requirements more difficult, and therefore the likelihood of building the

	wrong system increases. User participation increases the likelihood of capturing the right requirements.
3	**There are certain core activities for which user participation is especially important.** These include: (a) feasibility analysis, (b) information requirements determination, (c) defining input/output forms, (d) defining screen and report formats, and (e) the final installation of the system. The necessity of user participation in other activities is dependent on the complexity of the system, with more complex systems requiring more user participation.
4	**It is important to have people-oriented managers, especially if an organisation is still in the initiation stage of MIS use.** This type of manager is better at communicating with users in an environment where uncertainty and fear of change are high.
5	**It is particularly important to allow user participation by users that are functionally knowledgeable.** Users with functional expertise develop negative attitudes toward the system being developed if they feel they are being left out, i.e. if they have little or no influence over the development of the system. Users without this functional expertise are less likely to develop such negative feelings toward the system, even if they have no or only little input.
6	**There is an optimal level of user participation.** Though user involvement generally increases the likelihood of system success, increasing user involvement past certain levels may be counterproductive. Once users have contributed what they are best able to contribute to the development process, further involvement does not add value, but may rather be perceived as wasting time or resources.

2.4.4 Benefit to both users and developers

User involvement requires resources from users and developers, but it brings benefits to both stakeholders (Butz & Krüger, 2007; Shah & Robinson, 2007). Getting users involved in system development may also improve their attitudes toward the system, and involvement also increases the importance and relevance of the system perceived by the users (Lin & Shao, 2000). Early user involvement expands the understanding of users' values in addition to the fact that user involvement offers beneficial information about users' needs (Kujala, 2008). Hyysalo’s study on the development of a diabetes database management system showed that user–producer collaboration was significant to both parties (Hyysalo, 2010).

2.5 USABILITY AND USER PARTICIPATION IN HEALTH INFORMATICS

As previously stated, the focus of this research is in health informatics discipline. In more detail, usability and user participation in HIT system development is researched. As stated in the introduction, it is well known that the usability of the healthcare IT system is quite poor and users are dissatisfied with current tools (Alharthi et al, 2014; Lääveri, 2008c, 2008d; Smelcer et al., 2009; Vainionmäki et al., 2008; Walji et al., 2013).

In addition, Armijo, McDonnell, and Werner (2009) conclude that:

> *"… usability is often cited in the literature in relation to less than ideal results of EHR use, there is evidence that this issue is often poorly understood and is not adequately addressed by EHR developers and users alike".*

Armijo et al. present potential reasons why usability has not received the required level of attention and investment:

- EHR usability is **probably a more subjective and elusive concept** than identification of desired software features, functions, and interoperability goals.
- Clinical environment is difficult to replicate in laboratory settings and privacy concerns may **prevent using usability methods in clinical settings.**
- Health IT providers can be unable or **unwilling to invest largely in usability** (e.g. user acceptance testing, information design, and usability expert).
- The market is **unable or unwilling to pay** for usability-related development.
- There is a **lack of usability experts or EHR end-users** in EHR implementation teams (Armijo et al., 2009).

Accordingly, Jones (2013) has argued that user experience is not a significant business driver for healthcare information systems. Usability and user participation in health informatics is also processed in Paper I in the Introduction sections, Paper II sections 1.2 and 1.3, Paper III sections 1.1 and 1.3, and in the paper IV sections "Background on HITS development" and "User participation in IT system development".

2.5.1 User-centred design in healthcare IT system development

User-centred design aims to improve usability. According to Armijo, McDonnell, & Werner (2009), EHRs' usability improvements will not only support the care of the *"whole patient"*, but also improve the quality, safety, efficiency, and effectiveness of care delivered in the primary care setting. This claim may be generalised to other health care settings also.

Moreover, user-centred design methods have been proven to be effective in healthcare IT system development. According to Johnson et al. (2005), using user-centered design (also called usability methods) throughout the design life cycle helps to provide quality health care IT systems. Chan et al. (2011) have also proven that user-centred design can enhance task efficiency and usability in health care IT systems. Also, Sittig, Kuperman, and Fiskio (1999) conclude that usability methods help developers to understand how clinicians use computers and how systems can be improved. The positive effects of the user-centred approach when designing electronic patient records were also reported by Christensen et al. (2009).

Thursky and Mahemoff reported that UCD methodology called "Contextual Inquiry" was useful to describe the complex ICU (intensive care unit) work-environment, and also other usability methods, user case scenarios, and paper prototyping in parallel shortened the software development process (Thursky & Mahemoff, 2007). Also, designing and developing consumer health technologies with UCD methods (personas and user profiles) has achieved more positive experiences (LeRouge, Ma, Sneha, & Tolle, 2013).

Kushniruk et al. (2006) have successfully performed several usability tests to improve healthcare IT systems. They have stated that usability testing is

> *"a practical yet scientific approach to evaluating how usable our systems are and can also provide invaluable feedback to designers with ways of improving their usability, safety and work-flow".* (Kushniruk & Borycki, 2006)

Kushniruk & Patel presented a group of usability methods, which they have successfully used to evaluate clinical information systems (A. W. Kushniruk & Patel, 2004). The recently constructed 'EHR usability toolkit' presents methods and tools for measuring usability to support primary care providers in identifying usability issues in their current EHRs (Johnson et al., 2011).

User-centred design methods, e.g. observational techniques, provide a proper understanding of current realities and also put designers in touch with user reality and techniques, just as prototyping put users in touch with design (Heeks, 2006). In the next chapter, user participation in HIT development is discussed further.

2.5.2 User participation in health IT system development

Shah and Robinson have studied in which stages of (medical device) lifecycle users have been involved. The stages that they regarded were the concept, design, testing and trials, production, and deployment stages. The highest extent of overall user involvement was in the design phase, and the lowest in the concept stage. According to their study, the most commonly used methods for involving users were usability tests, interviews, and questionnaire surveys (Shah & Robinson, 2006).

Rahimi et al. (2014) argue that there is a positive connection between healthcare information system (HIS) success and early user participation. They also stress that healthcare managers and administrators must have a detailed plan for user participation early on; thus the users' real needs can be identified and verified and the users are committed to and motivated in the development (B. Rahimi et al., 2014). Høstgaard, Bertelsen, and Nøhr (2011) have also noticed the connection between real end-user participation and a successful outcome for HIT development. The positive effects of user participation in medical technology development, according to the literature review by Shah & Robinson (2007), were increased access to user needs and experiences, improvements in design and user interfaces, and improvements in the functionality, usability, and quality. Their review also pointed out some key impediments in involving users, which were lack of resources, communication and cooperation between users and developers, attitudes of technical developers, lack of understanding, and appropriate knowledge about methods to be used (Shah & Robinson, 2007).

It is not always straightforward that real end-users participate in the development process. According to Høstgaard et al. (2011), the balance of power and also the interests of the different social groups involved are decisive. Also, cooperative and communicative methods between IT professionals and the healthcare professionals in health IT projects need further development (Petersen, Bertelsen, & Bjørnes, 2013).

2.5.3 Knowledge Gap in usability and user participation in HI

In section 1.1 it was stated that health care information systems seem to encounter a lot of problems with usability. There is plenty of literature stating that user-centred design and user participation is an important part of software development. It has also been commonly argued that, through using user-centred methods and involving users in the development process, the usability of the software can be improved. This raises the question of why the usability of healthcare IT systems is poor, even if theoretical methods for its improvement are known.

3 Methodology

This study was carried out with both qualitative (interviews, action research) and quantitative (questionnaire survey) research methods. Therefore, the study can be called mixed method or multi-method research. Combining both kinds of methods aimed at triangulation.

Triangulation is a research strategy that aims at compensating the weaknesses of each single research method by studying the same phenomenon with different methods that have different benefits and weaknesses (Jick, 1979; Tashakkori & Teddlie, 2003). Integrating fieldwork and survey methods in the same study is the most common way to use triangulation (Jick, 1979). The reasons for selecting specific research methods for specific sub-studies of this research work are justified in sections 3.2-3.4 and in the individual Papers I-IV.

This chapter proceeds from the general to the specific. The philosophical research approach is presented first, in section 3.1. The research design and process are described in section 3.2, followed by descriptions of the specific data collection methods and data analysis methods used in the study, which are contained in sections 3.3 and 3.4, respectively.

3.1 RESEARCH APPROACH

The research philosophy and paradigm of the study are described in this section.

Research philosophy

This study as a whole is situated between three schools of thought in the philosophy of science: pragmatism, empiricism, and interpretivism (Figure 15).

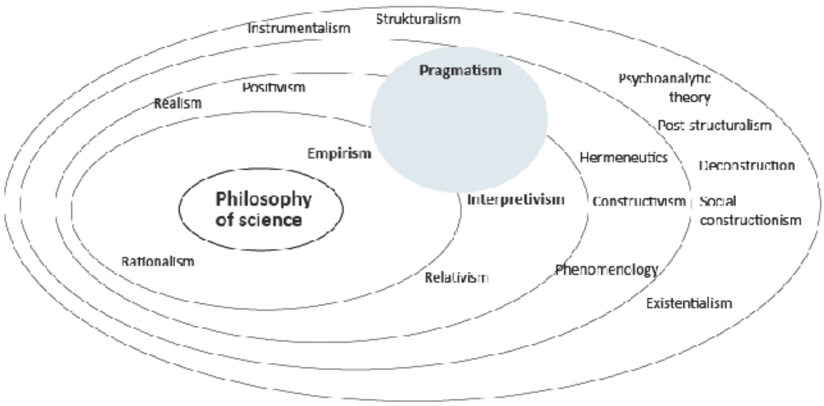


Figure 15. This research situated in the "map" of philosophy of science. Figure modified from (JYU, 2015)

Empiricism is a philosophy of science in which knowledge is based on observations and experience. Physical observations and empirical data are generalised to other research object-like phenomena. Rationalism and empiricism can be seen as two fundamental philosophies of science. *Interpretivism* emphasises interpretations and making interpretations in the production of scientific knowledge. It is a rather common philosophical background in qualitative research (JYU, 2015).

Pragmatism emphasises the practical nature of knowledge, as well as action and a practical orientation in conducting research, problem solving, and knowledge production (JYU, 2015). Pragmatism was formulated in the beginning of the 1900s by Peirce (1931), James (1907), Dewey (1931) and Mead (1938) to be a philosophic alternative to abstract and rationalistic science, having its foundation in empiricism (Goldkuhl, 2004).

"*The basis in human action gives pragmatism an orientation towards a prospective, not yet realised world [...] that pragmatism has an interest not only for what 'is', but also for what 'might be'. The basic interest for action in pragmatism is not conceiving action as an end in itself*" (Goldkuhl, 2004).

This study was conducted with a practical orientation; research questions were raised from practical concerns and from the needs of understanding and having better tools for solving practical problems in software development work. Thus it can be considered that pragmatism is the main philosophical approach applied in the study.

3.2 RESEARCH DESIGN AND PROCESS

Each sub-research question of Table 1 was investigated in a sub-study of its own. The research process and sub-studies are presented in Figure 16. Literature review and personal professional experience were applied throughout the process. The methods selected for each sub-study, as well as the materials and outcomes, are presented in Table 5. The actual outcome history and timeline from one sub-study and paper to the next is described in Chapter 4.

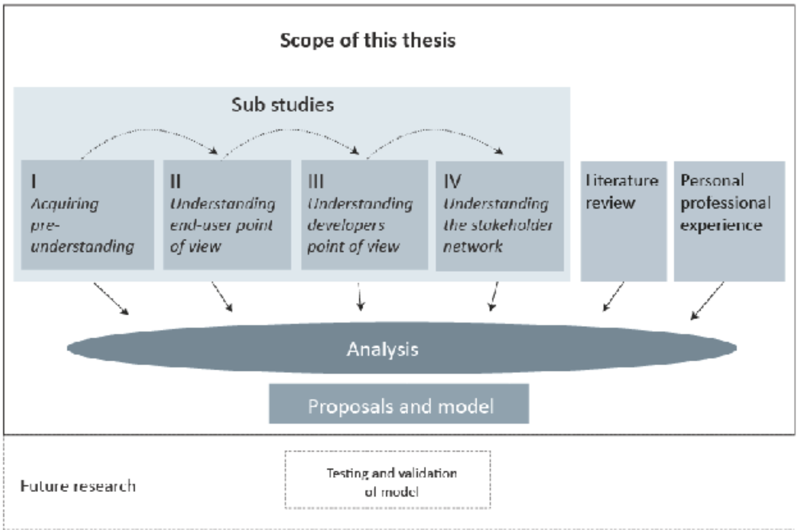


Figure 16. Research process and sub-studies

The research process began by addressing sub-research question 1 (SRQ1) to acquire a preliminary understanding. This was conducted as a case study in a pilot project of the ZipIT project (University of Kuopio, carried out in 2004-2007). In Figure 16 it is presented as sub-study I.

Table 5: The methods, materials and outcomes of the research question and sub-questions.

Research question		Methods	Data	Outcome
RQ	What factors impact the usability of HITS and user participation in HITS development in Finland?	Content analysis	All data from papers I-IV	Section 5.1
SRQ1	Are user-centred methods useful in HITS development?	Action research; observation, interviews, heuristic analysis, questionnaire	Observation notes, interview notes, heuristic analysis results, questionnaire answers	Paper I
SRQ2	What is the current state of the usability of HITS and user participation in HITS development from the user viewpoint?	Sub-study of a large national questionnaire study; statistical analysis of structured questions, content analysis of open-ended questions	3929 answers to questionnaire	Paper II

SRQ3	What is the current state of the usability of HITS and user participation in HITS development from the developer viewpoint?	Single-case study, questionnaire; same methods as SRQ2	136 answers to questionnaire	Paper III
SRQ4	Which issues in addition to user participation affect the usability development of HITS?	Expert interviews, content analysis	12 interviews, charts	Paper IV
SRQ5	What can be done to improve the usability of HITS and user participation in HITS development?	Logical reasoning	Data gathered in sub-studies I-IV, practical experience in HITS development	Section 5.2 and 5.3

SRQ2 and SRQ3 are both about the same current state, but from different viewpoints, so the same questions and methods should be applied in both sub-studies as much as possible. However, conducting a national questionnaire is not possible with one PhD researcher alone, so a single-case setting was selected for sub-study 3. The fact that the researcher is employed in one of the major HITS companies was used as an opportunity access all developers in the case company, as explained in section 4.4 and Paper III. This sub-study was carried out in 2013.

For sub-study 4 on SRQ4 regarding the broader landscape view, methods more explorative than a questionnaire were needed. Expert interviews were selected as the main method, using purposive sampling as described in section 4.5 and Paper IV.

Literature reviews were carried out constantly during the research process. All data collected in sub-studies I-IV and the literature reviews were used as input to an analysis phase, as described in Figure 16. Non-formal content analysis of the input was used for summarizing the answers to the overall research question in section 5.1 of the main thesis. Logical reasoning was then applied to address SRQ5 regarding ways of improving the current situation. This resulted in recommendations in section 5.2 and a holistic model in section 5.3.

During the research, the author has worked constantly in a healthcare IT company developing software. Thus the author was able to mirror the reality of the research process and findings. That enabled the author to broaden the perspective on the phenomenon studied in this research. The designer-professional experience and 'experiential knowledge' (e.g., EKSIG, 2015) were utilised to strengthen the study. Therefore, it can be argued that the study had aspects of action research, which is cyclical and aims to develop and change things in real life for the better. This research is inspired by action research, but as a whole it is not true action research because no intervention has been conducted in the extent specified for action research. Action research principles are presented in the following section.

3.3 DATA COLLECTION METHODS

The specific methods used in this study are discussed in the following subsections, providing arguments for where and how to apply each method.

3.3.1 Literature review

A literature review is conducted to have an understanding of the present knowledge and to know the research conducted in the field of usability, user participation, software development, and health informatics. According to Bhattacherjee (2012), the purposes of literature review are to **survey** the current state of knowledge, **identify** key authors, articles, theories, and findings, and **identify gaps** in knowledge in the area of the research interest.

The literature reviews in this research were not systematic reviews. Literature reviews were conducted mostly during summer 2013 – autumn 2015. For searching relevant information, the following tools have been used: Google Scholar, PubMed, Mendeley, Nelliportaali, and Google. The search terms that have been used include usability, user-centred design, user experience, health / medical information system, health / medical information system development, health care software, software process, software process improvement, user participation, user involvement, agile methods, and agile user-centred design integration, as well as combinations of the individual terms. Of course, when a research paper closely related to thesis topic was found, the list of references was reviewed and the articles that were most related to the topic of this thesis were chosen for further study.

3.3.2 Action research

Baskerville (1999) argues that in the literature there is a widespread agreement of four common characteristics of action research: 1) "an action and change orientation", 2) "a problem focus", 3) "an 'organic' process involving systematic and sometimes iterative stages", and lastly 4) "collaboration among participants." Myers (1997) states that the most popular definition of action research is Rapoport's:

> *"Action research aims to contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework."* (Rapoport, 1970 p.499)

An action researcher typically participates in the research "from the inside". The present state of the research on the topic and the things that affect it are typically defined first. After researching and modelling the present state, the researcher can concentrate on the future. The research is conducted by making interventions and following up the results of the interventions. Action research is cyclical and its goal is to develop and change things for the better.

In this study, the role of the researcher was mostly to observe and collect information about the present state. Interventions were made on a small scale only, affecting mostly the researcher's own work in the software industry. Sub-study 1, however, was conducted in an action research way but over only a short period of time.

3.3.3 Questionnaire

According to Bhattacherjee (2012), a questionnaire is *"a research instrument consisting of a set of questions (items) intended to capture responses from respondents in a standardized manner"*. There are two kinds of questions: unstructured, which the respondent can answer by free text, and structured, which offers a set of options to choose, for example in a Likert scale.

The questionnaire method was used in sub-studies 2 and 3 and in a small way in sub-study 1. The way in which the questionnaire method was used in sub-study 2 is described in Paper II. The strengths and weaknesses of using the questionnaire in that study are presented in the discussion chapter of the paper. Paper III section 2.1 presents how the questionnaire method was used in sub-study 3.

All the questionnaires were web-based. Vilkka argues that a web-based questionnaire is most suitable if all respondents are actors of the same organisation or company in which the population is large enough, giving them the same kind of technical possibilities for responding (Vilkka, 2005). That was the case in Paper III, while the respondent group in Paper II was larger and scattered throughout different organisations.

According to Tashakkori and Teddlie (2003) and Vilkka (2005), the common strengths of the questionnaire method are:

- Easy to get opinions of large and scattered groups of people
- Easy to ask about sensitive issues because the responses can be sent anonymously
- Perceived anonymity by respondents can be high
- Good for measuring attitudes and eliciting other content from research participants
- Inexpensive, especially mail questionnaire and group-administered questionnaires
- Can be administered to probability samples
- Quick turnaround
- Can be administered to groups
- Moderately high measurement validity for well-constructed and well tested questionnaires
- Easy data analysis for closed-ended items

In this study, the questionnaire method was selected to sub-studies 2 and 3 because it was a cost-effective way to get anonymous views of a large group of people sufficiently quickly, and also the data analysis of closed-ended question items was supposedly simple. Open-ended questions were supposed to benefit the research by adding respondents' real-life experiences and views, which could not be achieved by using structured questions only.

According to Tashakkori and Teddlie (2003) and Vilkka (2005), the common weaknesses of the questionnaire method are:

- Response rate can be low, especially for mail questionnaires
- When using web-based questionnaires, every respondent must have the same kind of technical possibility to answer
- Anonymity of respondents can be difficult to ensure in web-based questionnaires
- Needs validation
- Must be kept short
- Might have missing data or nonresponse to selective items
- Possible reactive effects (e.g., response sets, social desirability)
- Open-ended items:
    - possibly resulting in vague answers
    - possibly reflecting differences in verbal ability, obscuring the issues of interest
    - data analysis sometimes time-consuming

In this study it was observed that analysing the open-ended items was indeed a time-consuming task to complete. Formulating the questions and keeping the questionnaire short enough was challenging, as was making the questions easy enough to understand while getting the desired answers. Bhattacherjee (2012) points out the importance of designing questions: respondents must be able to read, understand, and respond to the questions in a reasonable way.

3.3.4 Semi-structured interviews

Qualitative interview is a powerful tool that is extensively used in Information Systems research. The term 'semi-structured' means that "*In an unstructured or semi-structured interview there is an incomplete script. The researcher may have prepared some questions beforehand, but there is a need for improvisation. The interviewer is the researcher or is one of a team*" (Myers & Newman, 2007).

The semi-structured interview method was used in sub-study 4 when collecting data by interviewing experts of healthcare IT system development. The interview method was selected because by using that method a deeper understanding of the phenomenon can be achieved.

The challenges of interviews are, according to Bhattacherjee (2012), interviews are time-consuming and resource-intensive; the interviewer needs special interviewing skills; and the interviewer herself or himself may cause bias in the responses.

Bhattacherjee also highlights some of the positive aspects of interviews: compared to a questionnaire, an interview is a more personalised method to collect data, and the interviewer has the opportunity to ask follow-up questions or ask the respondent to give more specific answers to clarify the answer (Bhattacherjee, 2012).

This method was selected to get a deeper understanding of the development network of healthcare IT systems and to enable the researcher to have an opportunity to ask the respondents to clarify their answers to questions.

3.4 RESEARCH MATERIALS AND DATA ANALYSIS METHODS

The materials of this research consist of the data of sub-studies 1-4 (Table 5). Sub-study 1 produced the most heterogeneous materials: data from observations, heuristic analysis, interviews, co-design, and a questionnaire. The sub-study 2 materials are 3929 responses to the large national questionnaire study, and the sub-study 3 materials consist of 136 responses to the single-case questionnaire study. The materials of sub-study 4 are 12 expert interviews as well as stakeholder charts drawn before and in between interviews. The materials are described on a more detailed level in each paper.

Two methods were used for data analysis: content analysis for interviews and open-ended questions, and statistical analysis for structured questions.

3.4.1 Statistical analysis

The statistical analysis of the quantitative questions of Papers II and III was conducted using the SPSS software and MS Excel software, and p-values (in paper II) were calculated using Chi-square tests. Further details are provided in the papers.

3.4.2 Content analysis

> *"Content analysis is the systematic analysis of the content of a text."* (Bhattacherjee, 2012)

Content analysis of the material in this study had the typical phases presented by Bhattacherjee (2012). First, the materials, i.e., free text answers of a questionnaire or interview notes, were read through, and the steps for dividing the text into segments were completed, leading to forming the initial categories. Secondly, the text was coded. According to Bhattacherjee (2012), *"For coding purposes, a coding scheme is used based on the themes the researcher is searching for or uncovers as she classifies the text"*. The last phase was the analysis of the coded texts to determine the most frequently occurring themes, to further specify the categories, and to summarise the data in each category. The tools used for content analysis were the MS Word and MS Excel software.

4 Summary of Papers

This chapter introduces the original research papers of this study. In total, four sub-studies were conducted in this research project and four papers were written. The continuum of the sub-studies and relations between them are presented next and in figure 17. The purpose is to guide the reader to what is essential in each paper for the whole of the thesis, not to pre-empt reading them.

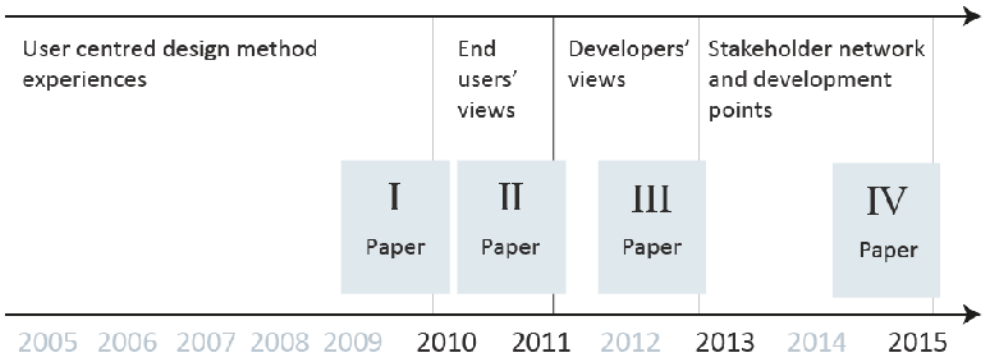


Figure 17. Research and papers in a timeline

4.1 RELATIONS BETWEEN THE RESEARCH PAPERS

The first paper described a study on experimenting with user centred and participatory methods in healthcare information system development, more precisely in the requirement analysis phase. The end-users who participated in this pilot project gave very positive feedback on the methods of user participation that were used. This sub-study was presented in the MEDINFO 2010 conference.

Because of the very positive feedback and the scarcity of knowledge on the methods used in user participation in healthcare IT systems development, a need arose for knowing users' opinions and experiences on HITS development on a wider scale. In the meantime, the Finnish Medical Association was designing a large questionnaire about healthcare IT systems to all its members (almost all Finnish medical doctors), a task that was carried out in 2010. The organisers of the questionnaire agreed to add some questions on the themes suggested by the author of this thesis. These questions were about the user involvement, development methods, evaluation of the current health IT systems, and visions on the future. The second paper was based on that questionnaire and published in the International

Journal of Medical Informatics. Papers on other parts of the large questionnaire were published by other researchers.

Thus, the second paper focused on discovering end-users', specifically medical doctors', opinions on and experiences with healthcare IT systems development and use, and on the medical doctors' participation in the design and development of healthcare IT systems.

The study about the opinions of physicians as end-users continued by investigating how the developers of those healthcare IT systems thought about end-user participation and the then-present state of HITS development. That became the basic idea for the third paper. The data collection for Paper III was conducted in 2013.

Sub-studies I, II, and III raised a need to understand what other factors affect HITS development. The fourth paper thus aimed at obtaining a wider view on the present state of HITS development in Finland. It first built up a landscape view of the stakeholder network in HITS development in Finland in order to identify the spots where development action is most acutely needed.

4.2 PAPER I: PARTICIPATORY INTERACTION DESIGN IN USER REQUIREMENTS SPECIFICATION IN HEALTHCARE

User involvement and user-centred design methods are widely seen as the key solution to usability problems. The first paper describes experiences of and development ideas for using user centred design methods in the healthcare IT context. The traditional user centred methods used were extended in this study.

The study was part of the ZipIT project, a research project on Activity-Driven Information Systems Development in healthcare (conducted in 2004-2007). The ZipIT project contained many pilot cases, and this study was conducted in a pilot called "Planning and following up medicinal care in hospital X".

The first thing to do was to find out and model the present state of medicinal care in the hospital. The present state was researched by visiting the hospital and following the end-users doing their everyday tasks, interviewing healthcare professionals, as well as participating in an end-user workgroup, doing artefact analysis, and modelling the information flows in medicinal care and the workflows of the end-users. A heuristic evaluation with end-user participation was also conducted on the existing medicinal care software.

Secondly it was necessary to model the goal state. The purpose of this was to design how the medicinal care in the hospital wards would be seen at a glance, how new prescriptions would be written, and how existing prescription would be changed. The goal state was modelled with the help of user interface drawings.

The user interface drawings, a kind of a low-fidelity prototype, enabled health professionals to describe their requirements and create a common understanding with the system developers. Traditional usability methods were extended in this study (Martikainen et al., 2010).

4.3 PAPER II: PHYSICIANS' EXPERIENCES OF PARTICIPATION IN HEALTHCARE IT DEVELOPMENT IN FINLAND: WILLING BUT NOT ABLE

The aim of Paper II was to study the present state of end-user participation and end-users' motivation to participate in healthcare IT system development. Another aim was to study what physicians thought about their IT systems at that moment and what kind of visions they have about future IT systems. These issues were studied with a web-based questionnaire as a part of a large national health IT systems study, which was responded to by 3929 Finnish working-age physicians. The response rate was 31.3% of all physicians in Finland.

The respondents were disappointed with the IT systems they used; the dissatisfaction was the highest among young physicians working in public hospitals. A lot of usability-related problems in the IT systems in use were highlighted in the responses to the questionnaire.

The four questions of the large questionnaire that were dealt with in Paper II focused on user-oriented IT development. The results were divided into three parts:

1) Physicians' experiences on giving feedback and on development activities in general.
Feedback-giving and development activities, seen as the collaborative activities between the developers and end-users of a software system, need re-thinking – significant challenges were identified from the physicians' responses.

2) Physicians' interest in participating in and contributing to the development activities.
Physicians seemed to be motivated and interested in contributing to and participating in the development of their software tools. In addition to the multi-choice options given in the questionnaire, other ways of participating were also suggested.

3) Open-ended comments and visions of future IT systems.
In summary, the physicians' view of current IT development activities was that developers never ask users' opinions and instead real end-users need to participate more in the development and testing of healthcare IT systems.

Physicians' experiences in participating in healthcare IT system development were frustrating – feedback on proposals was never obtained from the developer companies, but still IT systems development work would be easier if end-users could have specific time to do that.

Some really necessary functionalities were regarded missing, such as a proper 'observation chart'. Respondents also noticed that they could not see the big picture of a patient's condition at a glance, and navigation in the system was not unambiguous. Several usability-related errors were brought up. A few positive experiences in the use of the respondents' healthcare IT systems were brought up, too.

In the answers to the question about the future vision of healthcare systems, numerous respondents highlighted that the system used should simply work as expected, without errors or technical problems, and be as reliable and trustworthy a tool as a pen and paper. A few descriptions of a future goal state were presented. It was made clear that significant attention is needed to make improvements in user experience.

A lack of user-centredness in healthcare IT systems development was emphasised in this study. The findings indicated the need for enhanced interaction between developers and end-users and the need to improve the methods and practices for participatory healthcare IT development. This sub-study enabled the researcher to create a comprehensive and in-depth understanding of the present state of HITS development, the problems experienced, and the collaborative experiences of physicians (Martikainen, Viitanen, Korpela, & Lääveri, 2012).

4.4 PAPER III: USER PARTICIPATION IN HEALTHCARE IT DEVELOPMENT: A DEVELOPERS' VIEWPOINT IN FINLAND

Paper III presents the results of a web-based questionnaire that was conducted in one of the major HITS provider companies in Finland about developers' views and experiences in healthcare software development. The aim of this sub-study was to find out about the following questions: What is the developers' view on end-user participation in HITS development at the moment? How would developers want end-users to participate in systems development? Do the developers' views differ from the physicians' (end-users') views of the current state of collaboration in developing IT systems?

The respondents of this questionnaire were mainly experienced workers and had worked with users. The response rate was good at 37%. In total, 9 questions, both qualitative and quantitative, were included in the questionnaire. Two of the main questions were adopted from sub-study that was presented in Paper II: a question about user participation and a question about experiences in users' feedback and development activities. The third main question dealt with the software process and methods used.

The findings were compared to the study presented in Paper II. Between the two respondent groups (physicians and developers), it was a common opinion that the pace of implementing corrections and modifications was dissatisfactory. Most of the developers thought that they were interested in user feedback and took the end-users' opinions and experiences into account when developing software.

The most popular means of user participation, among the respondents, were that 'users would present their work and needs related to it in their workplace'. The second most popular was user groups. The developers also suggested many traditional user-centred and usability design methods for use in HITS development activities.

In conclusion, it seems that both parties, end-users and developers, are interested but somehow not able to collaborate in a successful way. Possible reasons for the differences in views include the fact that there is no return channel of communication on what happened to the end-users' feedback, and that developers collaborate with customer representatives who are not end-users. It is obvious that there are one or more places along the route between the "end developers" and end users where there is a breakdown of the information flow (Martikainen et al., 2014).

4.5 PAPER IV: WHERE DOES THE INTERACTION BREAK DOWN? THE STAKEHOLDER MAP OF HEALTH IT SYSTEMS DEVELOPMENT AND USE IN FINLAND

As sub-studies 2 and 3 revealed, the developers of healthcare IT systems strongly argue that they do work with users and are highly interested in their views and take them into account, but the end-users of the same systems claim the opposite. Thus the question that needed to be addressed with the forth sub-sub study was: 'If both sides want to interact with each other, but the user experience is still not good at all, where does the interaction break down?'

The aim was to expand from the previously studied viewpoints of end-users and developers to the whole development network of healthcare IT systems in Finland. To achieve the goal, expert interviews were conducted. The interviewees were selected with a purposive sampling method.

On the basis of literature and prior knowledge, an initial map of stakeholders was drawn. In total 12 interviews were then conducted in two phases: the first phase of 5 interviews was to ensure that relevant questions were included, and the second phase of 7 interviews was to complement the material. The initial map was developed further during the interview periods. The interview materials were analysed with the content analysis method.

As a result, the initial map was enriched with new stakeholders and interactions between them. The content analysis also produced 7 categories of problem areas, interpreted as development spots. Most of the development spots related to software providers' and healthcare providers' interactions and collaboration. The following categories were formed:

1. End-user – developer collaboration;
2. Healthcare organisation – software provider collaboration;
3. Clinical knowledge and end-user participation in development activities;
4. National authorities;
5. Internal processes of the stakeholder organisations;
6. Research-based, evidence-based design and assessment;
7. General issues of HITS development.

This sub-study revealed that it is not only the software product itself (whether with good or bad usability), but also several other factors that may affect user experience. If the goal is good user experience, all of the other factors need to work together seamlessly. The main contribution of this sub-study was the description of the complex development network of HITS in Finland in the form of a map, in a birds-eye and multi-professional view. Such a view had previously been "tacit information" among experts, but was not presented in the research field. Identifying the development spots where problems in the development of healthcare software products are likely to occur and understanding what kinds of problems typically come up is a very important step towards improving the products, practices, skills, conditions, and other factors that hinder the positive end-user experience of healthcare IT systems. (Martikainen, Korpela, Luukkonen, & Vainikainen, 2015)

5 Results

In this chapter, the results of the study are presented. Problems need to be identified in order to develop the software development methods that will help produce better, more usable software for healthcare workers. Firstly, the findings arising from the four sub-studies are summarised and linked to literature in section 5.1. Secondly, in section 5.2, recommendations are made on the basis of the findings and the researcher's practical experience in healthcare software development. A new enriched model for HITS development is then outlined in section 5.3. Finally, section 5.4 contains the author's own reflections on the results.

5.1 FINDINGS FROM THE PAPERS: PROBLEM AREAS

In this section, the findings on problems in HITS development from the user and usability point of view are summarised. The full findings were originally presented in the four sub-studies presented in Papers I-IV.

Findings were reprocessed paper by paper, and in total 17 categories were formed. As shown in Table 6, most of the findings are from sub-studies II and IV. The re-processing of the research data was carried out from a new perspective. The goal was to produce more specific problem categories from the entire research as a starting point for developing proposals to address the problems.

Table 6: Number of findings from each sub-study (paper)

Finding from	Number of problem categories
Sub-study I	5
Sub-study II	11
Sub-study III	6
Sub-study IV	14

Sub-study IV resulted in a map of the stakeholders of HITS development in Finland (Paper IV, Figure 6), copied here as Figure 18 for readers' convenience. In addition to the identified stakeholders, it presents the main problem areas identified by the interviewees, interpreted as development spots (lightning symbols with numbers). The numbered development spots are described in the original article.

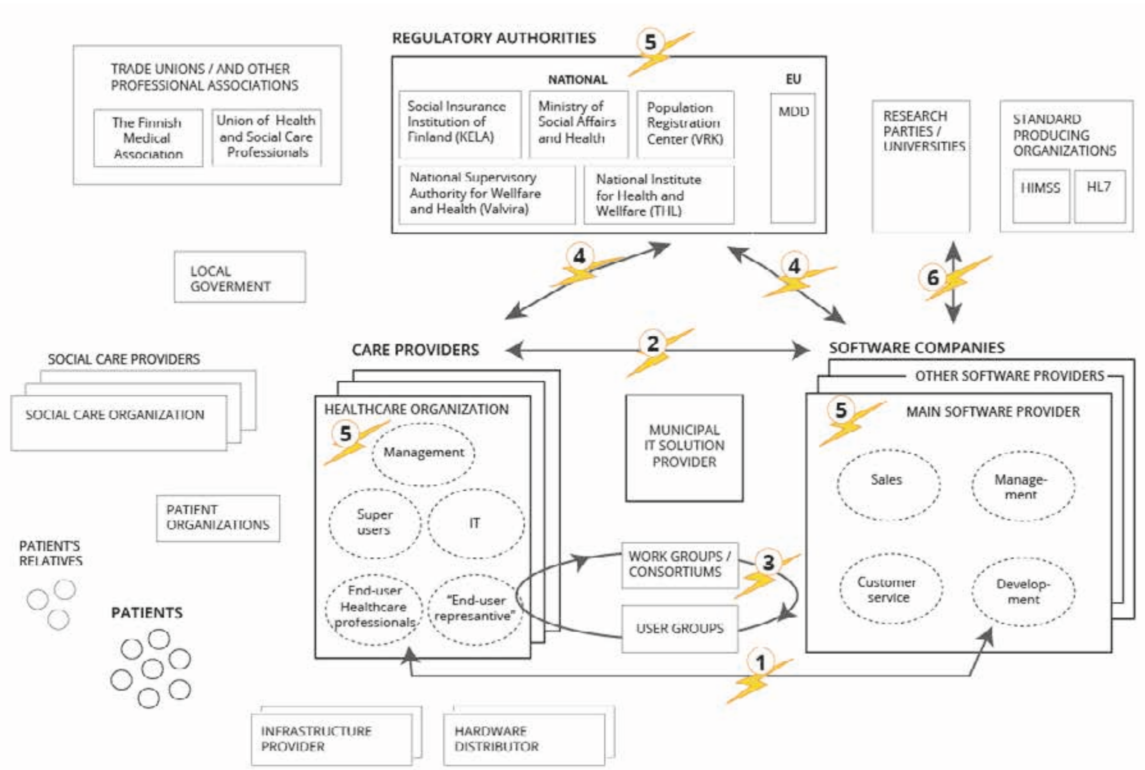


Figure 18. Stakeholder map and problem areas from paper IV.

All the findings presented in this section could be placed on the map as well. Findings presented in this section are naturally partly the same as those in paper IV. This section combines all the data collected in all sub-studies I-IV in order to provide a larger and more comprehensive contribution.

The categories are presented in the sub-sections below. The header of each sub-section describes the found problem area on a general level, and the numbers in parenthesis at the end of the header indicate in which sub-studies the problems in this category were found.

5.1.1 Domain knowledge of healthcare (I, II, III, and IV)

The results of every sub-study stressed the importance of the understanding of and knowledge about the healthcare domain when developing healthcare IT systems, and brought up the inadequacy of such knowledge currently. Domain knowledge was obviously seen as a mandatory precondition in order to develop appropriate tools for patient care. Software developers need a better understanding of the end-user work processes, information needs, and working environment. End-user physicians even argued that their point of view was missing in IT development, and healthcare IT systems were designed and developed by people lacking knowhow of the substance.

End-users' working processes, patients' processes, and the crossing points between them need to be understood in depth. Physicians and nurses are normally repeating some piece of the patient process throughout their workday with different patients (e.g., a physician appointment in a health centre), and a patient's process intersects several healthcare workers' workflows. For instance, a patient visit in a health centre includes:

1. time booking: nurse ->
2. appointment: doctor ->
3. laboratory test: sampler ->
4. laboratory analysis: analyst ->
5. x-ray: nurse ->
6. x-ray opinion: radiologist ->
7. appointment: doctor).

The software must support the workflows of the healthcare workers as well as the patient processes, for instance, to ensure efficient resource usage in the healthcare organisation.

The healthcare worker needs to access the relevant information in the right situation in the right format. To better understand what information needs to be seen when and how, we need a better understanding of the users' work tasks and workflows and the information they need in various care situations. Butz and Krüger (2007), too, stress the need for looking closely at the processes in the hospital environment when designing user interfaces.

In medicine and healthcare, there are a lot of specialties: different specialties mean different workflows and requirements for the tools. A HITS that fits all might be unreachable. Understanding the varying needs of practitioners in different medical fields is important. When trying to cover and please too many specialties, such compromises might be brought about that might not be satisfactory for any of the groups.

Also the lack of understanding of a user organisation's policies and activities was regarded as problematic when developing HITS. Knowledge is needed about how the user organisation operates and the hierarchy and allocation of responsibilities it has.

Smelcer et al. (2009) have also argued that because there are many specialties and the specialists have different needs, designing usability is complicated. In a study by Ferreira et al. (2008), it was noticed that healthcare professionals as end-users did not participate in designing and implementing the HITS. As a result, their needs were not considered enough and developers did not understand their needs, and therefore at the end the HITS did not reflect their workflows.

5.1.2 Developers' interest on users' wishes and experiences (II, III)

Physicians feel that software providers are not interested in the end-users' feedback and experiences; more than 60% of the physician respondents disagreed with the following statements: *"Software providers are interested in end-users' feedback"*, *"Software providers are cognizant of end-users' experiences and opinions"* (Paper II).

IT developers never ask for the users' opinions or experiences and end-user viewpoint is ignored — this kind of experience seemed to be dominant. In the physicians' opinion, developers seemed not to be at all interested in the users' needs or visions and almost never visited physician workplaces at hospitals or health centres to see realistic workflows and conditions.

Some of the differences between the opinions of physicians and developers are striking. About 90% of the developers thought that they were interested in the feedback given by users, while only 13% of the physicians viewed developers as being interested in their feedback. Also, when asked about taking end-users' opinions and experiences into account, more than four in five (81%) of the developers agreed, but only one in ten (9.8%) of the physicians agreed (Paper II; Paper III).

In summary, the respondent physicians wished that their needs, opinions, and experiences were better acknowledged and appreciated; on the contrary, the respondent developers argued they were indeed interested in user wishes, needs, opinions, and experiences. This problem did not emerge in sub-study IV, but can be placed on the map (Figure 18) to the direct relation between end-users and developers (problem area/lightning symbol 1).

5.1.3 User participation in general (I, II)

User participation is very important, but almost as important is the way in which the users are involved. Currently some of the users are very frustrated; they have tried to contribute to the development, but have already given up and are no longer interested. It was also noted that, in Finland, the procurement of healthcare IT systems is regulated by laws and rules that do not sufficiently promote participation and communication between end-users and developers.

Boivie et al. (2006) emphasise that one of the obstacles to usability and user-centred approaches in system development is the powerlessness of users in systems development, which means that users cannot often influence the actual design. User participation has proved to have a positive effect on the satisfaction of developers (Subramanyam et al., 2010). It is also already known that end-user participation is essential in order to have successful healthcare IT systems (e.g. Høstgaard et al., 2011; Shah & Robinson, 2006), and a detailed plan for user participation is recommended (B. Rahimi et al., 2014).

5.1.4 User participation methods (I, II, III and IV)

In questionnaire sub-studies II and III, five options were presented about how user participation should be organised. One of the options was that users would present their work and needs to developers at their workplaces, which was the most popular choice among developers and second most popular among physicians. In sub-study IV, interviews revealed that the collaboration between end-users and software developers seems to work quite well as long as the developers visit end-users and as long as the right end-users participate. It was also stressed that, when visiting end-users, developers like to interact with those users who could represent more than their own perspective only, having knowledge of other workers' activities also.

Physicians wished to share their thoughts with their colleagues: having "a physician responsible for collaborative activities with the software provider" was the most popular option in the questionnaire among physicians. Among developers, this option was the second least popular.

User groups was the second most popular option among developers, but physicians had the opposite opinion of that user participation method. User groups are currently used quite often as a method for user participation and gathering user needs and requirements. The HITS developers seem to wish to continue this trend, but most of the physicians disagree. Physicians had been asked to participate in a user group arranged by the company providing their HITS in order to contribute to the software development, but physicians found user groups to be pointless. One respondent even commented that this kind of participation was a waste of time.

The least favourite option among developers was direct feedback to the developers, for example by email, but this option was the third most popular among physicians. Sending feedback and development ideas to a web-based forum got some support from both groups.

Users participated in HITS development by testing software. It was emphasised that the suitability for clinical use must be confirmed by testing software carefully before deploying it. Besides, physicians were not always sure why updates or new versions had been installed, and in their opinions, new versions of the existing software were not always better than the old ones.

From the results, it seems that continuous usability testing throughout the design process is needed to ensure the usability of the software. Choosing the right user to test the product during the development phases is important. To get valid results, the test users should not be the same people who participated in the design and development, because the latter already know too many of the compromises and decisions made.

According to Hyysalo (2006), product developers are unfamiliar with the methods needed to design the use of a product.

5.1.5 End user representatives (IV)

It is usually necessary to choose some of the users to represent all end-users in HITS development. Choosing the right people to represent end-users is challenging - the level of users' knowhow varies significantly. Some users participate alongside their clinical work, while others may have left the clinical work a while ago, for example, due to their responsibilities in the organisation's IT department.

It seems obvious that clinical expertise should come from the specific field for which the application is intended. However, that might not be the case always: in sub-study IV, some considerations emerged about the role of user representatives. "Is the end-user representative participating in development able to represent the various kinds of end-users sufficiently? Is it even feasible to expect an end-user representative to be accountable for safeguarding all kinds of end-users' needs (including physicians and nurses from various different medical specialties) when representing clinicians in software development workshops?" (Paper IV). As a consequence, end-user representatives' roles should be as intermediaries and mediators between end-users and developers.

5.1.6 Users are not designers (I, II)

Listening to the end-users is extremely important when developing tools for them. However, it is crucial to keep in mind that users are not designers and should not be considered to be representing or replacing the persons who are specialists in designing human-computer interactions and user interfaces. Users typically point out problems by presenting some ideas, but their design solutions might be insufficient as the starting point for design. A need for user interface and usability design specialists was raised in sub studies. Also, Kaipio (2011) and Nies and Pelayo (2010) have stressed that when developing ICT systems with high usability, usability-skilled designers are needed. In addition, a human factors specialist may bring benefit *"in terms of reduction of (i) the number of iterative developments and (ii) the users' training costs"* (Nies & Pelayo, 2010).

Instead, end-users are experts in medical or nursing practice and their own clinical work, and therefore are the best to determine the kinds of problems that exist with the currently used IT systems. When designing HITS usability and user interfaces, ideas from other IT industries (for example from the game industry) should be utilised. To conclude, end-users are the most important and primary source of use-related information, not designers.

5.1.7 Clinical knowledge in development (II, IV)

Some of the questionnaire respondents (II) and interviewed experts (IV) argued that there are not enough experienced clinical physicians involved in the software development, and that healthcare IT systems are developed by engineers and those medical doctors who work in administrative positions. It is crucial to have current clinical experience in software organisations; people who have up-to-date clinical knowledge and end-users expertise should be a standard part of software development (e.g. in requirements specification, clinical testing, and prototyping).

5.1.8 Changes, improvements and feedback channel (II, III, and IV)

Under ten percent of the physician respondents agreed with the following statement: "*Software providers implement modifications as requested*" (Paper II, Table 5).

However, slightly more than half (55%) of the software provider's employees thought that customers' proposals for corrections and modifications were being implemented as requested (Paper III, table 3).

Even though physicians complained or made proposals for corrections or new features to their current system, it seemed to them that complaining did not change matters and existing errors were not fixed. In physicians' opinions, developers quite often rejected change requests presented by physicians by referring to technical reasons. Most often, no feedback about a proposal or correction request was acted on, which caused frustration among end-users. End-users rarely or never received news on what happened to the feedback they gave, because there seemed to be no two-way channel of communication. The software development process looks like a black box to end-users. No one seems to know what will ultimately happen to end-users' wishes. Some physicians wondered if end-users' wishes might be buried under a pile of "to-do" tasks or possibly just sent directly to the trash can. End-users are uncertain if the features they suggested would be added in a new version, if a reported problem would be corrected in the software and, if it was corrected, when the changes would be made. From the viewpoint of the end-users and the healthcare organisation, getting feedback about the reported problems was troublesome.

There are too many participants along the route between "end developers" and end users (Figure 18). The interviews for Paper IV revealed that the information flow from end-user to developer may contain as many as five intermediary steps:

1. end-user ->
2. hospital or office IT support person ->
3. municipal IT company help desk ->
4. software provider's customer support ->
5. product manager -> software developer.

End-users are not sure if their complaints are ever heard by the right people in the software company.

Users and developers have conflicting views on how changes and improvements to existing products are made. Somehow, it seems that both physicians and developers are "willing but not able" to collaborate with each other. Possible reasons for mismatches between users' and developers' views are discussed in Paper III, section 4.1.

Petersen et al. (2013) have also argued that there is a need to develop a better understanding of the HITS development process from the user point of view.

5.1.9 Holistic development (I, IV)

Developing HITS requires a really good understanding of the entire picture of system use. More holistic development needs to be emphasised. It is not enough to only see the software that is currently under development because end-users must work with many software systems during their working hours and these systems should be well integrated, both regarding technical and "logical" integration, to support the users' work.

According to the interviewees for Paper IV, at the moment only a tiny fraction of functionalities are designed and implemented at the same time, and due to this fragmented development, it is difficult for all the pieces to work together.

Furthermore, it should be kept in mind when developing HITS, at least in Finland, that healthcare and social care are required to work together more and more.

5.1.10 Lack of shared understanding of requirements and development goals (IV)

Shared understanding is a necessity for the project to succeed, but the experts interviewed for Paper IV explained that somehow the healthcare organisation and the software provider, or end-users and developers, do not have a shared understanding of the goals and concepts in the beginning of the development of new functionality and versions. Specification documents are difficult to understand in a shared way, and prototyping was identified as an excellent method for communication and development and for obtaining a shared understanding.

Prototyping has been recognised to be an efficient tool in user involvement, for example, so that the users evaluated the prototypes as a part of their daily work (Pekkola et al., 2006). Prototyping has also been identified to be an effective tool when collecting requirements, according to Boehm (1996):

> *"A prototype is worth 100,000 words. Written requirements specifications trying to describe the look and feel of a user interface were nowhere near as effective as a user-interface prototype".*

5.1.11 Simultaneous development of work and software (II, IV)

Developing software and work activities at the same time is an important issue in healthcare IT system development, but for some reason it is currently not a common practice. In sub-study IV, interviewees stated that the simultaneous development and re-organising of organisational processes, care processes, and IT systems is too rare. It is challenging work, but it is important in order to achieve well-functioning practices. One challenge is that the healthcare organisation and the software provider have different development schedules; it can be difficult to schedule collaborative development activities in such a way that fits both stakeholders' development goals.

The need for simultaneous development was already recognised by Toivanen et al. (2007), but it seems that the idea has not properly taken place in practice.

5.1.12 Development goals and prioritising (IV)

Currently, there are considerable numbers of development needs in healthcare organisations, and prioritising them is challenging. However, requirements set by regulatory authorities often take precedence. Prioritising other needs and requirements is complicated – whose wish is the most important? Could end-users be more involved in deciding which wish is implemented?

5.1.13 Time spent on development (II, III, and IV)

Under ten percent of the physician respondents agreed with the statement *"Modifications are implemented within a sufficiently short period of time"* (Paper II, Table 5). After a long development period, users do not seem to be satisfied with the results. It raises a question of whether enough end-user testing is done when corrections are implemented, or if the right persons accept the corrections. Likewise, only one fifth of developer respondents in sub-study III were of the opinion that modifications are implemented in sufficiently short a period of time.

Accordingly, the time used for development activities was too long in the opinions of both respondent groups. Therefore, developers end up working with quite different products than the end-users, who are struggling with existing software problems. It appears that developers and end-users are *"in totally different worlds, as if in different time zones"* (Paper IV). The long period of time may influence the users' view – they may think their feedback is not considered in software development activities, when in fact the developers are fixing the issues as quickly as possible.

5.1.14 Evidence based HITS development (II, IV)

Physicians emphasised a need for common standards for healthcare IT systems, and that IT systems should be consistent and interconnected (Paper II, table 2). More research knowledge should be utilised in HITS development activities. Just as medicine is evidence-based, the development of tools for medical practices should also be evidence-based. The software development industry would benefit from and should collaborate more with health informatics and usability research, both from the results of the research (e.g. user interface structure and interaction design) and the research activities themselves (e.g. assessment of currently used HITS).

Five research-based critical development spots for health IT systems are presented in a recent dissertation (Kaipio, 2011): development of efficient and mobile documentation; redesign of EHR user interfaces to streamline interaction sequences; ICT solutions to support communication and collaboration; customisable and context-specific IT systems; and conceptual redesign of nursing documentation systems.

A recent evaluation study of physician satisfaction with EMRs also suggests that continuing evaluation should guide the future selection and introduction of EMR systems (Alharthi et al., 2014).

5.1.15 Software provider organisation: Work practises (III, IV)

The developer respondents had positive opinions on all three statements about the present state of software development within the question about the software development process and methods used (Paper III, Figure 2). However, the open comments were more negative.

About half of the developer respondents agreed, while more than one third disagreed, to the statement: "In my experience, the feedback from customers will be communicated from 'customer interface' to software developers". Almost three quarters of the respondents agreed with the statement "I work with users". The statement "The current way of developing software supports co-operation with customers" divided the opinions the most. Only 42% agreed that the process supported customer cooperation, while 28% disagreed, and about the same amount (30%) had a neutral opinion on that statement. User participation was seen as important, but no specific ideas were presented on how to improve the collaboration with users (Paper III).

The developers stressed the need for resources for more intensive collaboration with end-users, such as focusing on analysing and understanding user feedback. They also stressed the need to develop further collaboration with other professionals (such as customer support, development, sales) inside the organisation in order to avoid misunderstandings (e.g. a salesperson should not promise with the customer to deliver features that are impossible to implement).

IT procurement would benefit if people capable of affecting software application development (e.g., people like software architects), were also involved, not just people who are essentially salespersons from the healthcare organisation's point of view. Also, it is important that the right software development professionals participate in the right phases early enough (e.g. usability designers in the requirements phase).

It has been noted that it is challenging to develop new features based on the ever-changing requirements of healthcare organisations, especially within existing software that in some cases has been built with obsolete technology.

5.1.16 Healthcare organisation: Work practises (II, IV)

Physicians had quite negative experiences of collaborative activities between themselves and people in managerial positions in healthcare organisations (Paper II, Fig. 1). Nearly half of the physician respondents disagreed with the statements *"When I want to give feedback I know to whom and how I can send it"* and *"In our organisation, people in managerial positions are interested in end-users' experiences and opinions about the used IT systems"* (Paper II).

Respondents argued that healthcare organisations should give more opportunities to busy clinical staff to participate in IT development; participating in IT development would be more motivating if it could be done during the official working hours, not as extra work.

Even if many of the healthcare organisations' IT staff have a background in a healthcare profession, the professional jargons differ, and the "lack of a common language" makes communication between end-users and IT staff challenging.

5.1.17 National authorities (IV)

Experts interviewed for Paper IV were dubious about whether the national authorities understand current software development practices and have enough clinical experts participating in the national requirement specification process. National authorities were expected to be more agile when making national level definitions and specifications. End-user participation in this national requirement engineering was regarded to be very important.

The national requirements that end-users record large amounts of information have been criticised. The usability of clinical software systems has suffered because of the level of detail that was required to be recorded in the systems. Other research has also shown structured data entry interfaces bringing usability challenges to end-users (Walji et al., 2013). However, structured data interfaces have positive effects too:

> *"More advanced use of documentation templates led to greater opportunities for improving quality of care. For example, problem-specific templates (such as a sore throat template) with embedded prompts reminded clinicians to ask about particular symptoms, order particular tests and prescriptions, or perform preventive or disease management activities. Also, templates that help clinicians enter data in coded rather*

than free-text form facilitated more advanced computer-based decision support for such tasks as care coordination and chronic disease management" (Miller & Sim, 2004).

Mäkelä (2006) argues that the more structured the information is, the easier it is to structure, analyse, and automatically process on a computer, and the easier it is to exchange information between software systems. He also confirms the outcome of this study of the usability challenges of structured data - entering data is more difficult and slows down if everything is too structured (Mäkelä, 2006).

In paper IV, interviewed experts argued that the specifications from national authorities are not in line with users' needs and requirements. It was also argued that the ordinary end-user is not familiar enough with issues related to the new national electronic archive of patient records (KANTA).

5.2 RECOMMENDATIONS BASED ON THE FINDINGS

The findings presented in the previous section were analysed in light of the researcher's own experience in HITS development to formulate recommendations for the usability development of HITS. The resulting nine recommendations are presented and mapped as software development phases in Table 7.

Some of the recommendations relate generally to the whole development process. Some are more specific, to be utilised in a specific activity only, such as requirement analysis, software design and implementation, or validation and maintenance. The recommendations are intended to be applicable to all stakeholders in the HITS development network. Some of them deal more with the inner activities of one type of organisation only, while some deal with the collaboration between different stakeholders.

Table 7 : Recommendations based on the findings

Software development phase	Recommendation	Findings of section 5.1
Throughout the development	1. Develop the end-users' work practice and the software simultaneously	11
	2. Increase developers' domain knowledge	1, 7, 17
	3. Identify stakeholders that impact and have responsibility to design the good usability of software	9
	4. Make the activities of the software development process more visible to different stakeholders	9, 17, 13, 8, 2, 12
Requirement analysis	5. Plan user participation and carefully choose the users who participate	17, 16, 15, 12, 5, 6, 3, 4
	6. Get a holistic big picture and model the development target to achieve a shared understanding	10, 9

Software design and implementation	7. Develop the HITS in close collaboration with users	7
	8. Develop evidence-based design patterns for HITS user interfaces and interaction	14
Validation and maintenance	9. Inform users about changes and why the changes are made, and provide feedback to users about their 'change requests' and 'development wishes'	8, 17

5.2.1 Develop the end users' work practice and the software simultaneously

Introducing a new software system may always require some level of change in the work practises of the users. If the work flows and practises are developed simultaneously with the software, the development will result in efficient and seamlessly interoperable new ways of work and tools of work. At that point, the total usability experienced by the users will presumably be better. If different groups of people are involved in the development, it should be agreed upon in the beginning how the co-development (collaboration between different developers) will be organised. The important point is that the software and the practises and processes are developed simultaneously through the development process. Ferreira, Cruz-Correia, and Antunes (2011) argue that the main reason for HIS failure is that, because healthcare professionals do not participate in the design, they need to adapt their workflows around the systems. In the ZipIT research project, which was the starting point for this thesis research and in which research data for the paper I was gathered, the need for simultaneous development of the information system and work improvement was raised (Toivanen et al., 2007).

5.2.2 Increase developers' domain knowledge

It is highlighted in the health informatics research that developers need to achieve better healthcare domain knowledge (e.g. Kaipio, 2011), but how exactly this could be accomplished has not been given much attention. Based on this thesis, few means can be recommended to increase developers' domain knowledge. Developers should visit end users' working places, observing them while they are doing their patient work or other typical tasks.

Clinical specialists, physicians, nurses, or clerks with fresh practical work experience are also extremely valuable in the development unit or team to bring the users' mind-set and working reality closer to the developers.

5.2.3 Identify stakeholders that impact and have responsibility to design the good usability of software

The stakeholder map of HITS development in Finland in Paper IV aims at depicting how many actors' collaboration it takes to develop software for healthcare. Usability should be every stakeholder's goal, not only the usability designer's concern. If the viewpoint is usability-centred on all levels, the design process should lead to good outcomes from the end-user perspective. It would help to develop usability if more actors than only the usability designers had more understanding of UCD methods, for example, how they affect one's own job.

Usability experts come from various backgrounds. Some of them are clearly graphic designers, some are interaction / usability designers, and some focus on front-end development. Who ultimately decides on usability, and the designer's role in the development process, are sometimes a bit ambiguous.

It is important to identify in which issues compromises can be made, and if the possible conservative attitudes of development team members restrict the development of usability.

5.2.4 Make the activities of the software development process more visible to different stakeholders

Previous studies have rarely emphasised that the software development process should be made more transparent for other stakeholders. (Petersen et al., (2013) argued that: "*there is a need to explore new ways of doing system development that enables the clinical practice and the system development practice to meet in a shared language-game and design space – a game that also enables the project manager to handle the project*".It is important to identify what software development requires from end-users, and what from developers.

Additionally, it is helpful for end-users to understand why software development is taking such a long time, especially since both developers and end-users thought that developing software takes too long a period of time (papers II and III).

Users probably do not know what software development demands, and certainly one can wonder why they should. However, if end-users had a better understanding of the complexities involved in developing software for healthcare (because it necessitates the contributions of so many different professionals and even multi-professional collaboration between different organisations), maybe it would help them understand that the development phase requires a significant amount of time and resources, and even contributions by the end-users themselves.

It would also be good to bring to light how wide a variety of development proposals, sometimes conflicting with each other, the end-users themselves provide. Maybe end-users could resolve between themselves whose proposition would be the best for all, or what kind of configurability characteristics the software should have, so that the needs of the heterogeneous user groups could be better satisfied.

5.2.5 Plan user participation and carefully choose the users who participate

Several studies have confirmed the importance of user participation in HIT development (e.g. Høstgaard et al., 2011; B. Rahimi et al., 2014; Shah & Robinson, 2007). That alone is not enough to improve the usability of HITS. There is a need to decide which of the users should participate, and when and how. Therefore, it is necessary to distinguish between the end-user representatives' role and the actual end-users' role in development tasks.

The plans and prototypes of the software do not need to be too "polished", but rather the end-users should be involved at a very early stage to ensure that the direction is right. The developers, however, need to have a strong view of what is being developed, since users can give very contradictory comments, particularly if the plans are quite radical.

A reformist attitude is important both to the users and to the developers, since innovations are born if it is possible to break away from established ways of thinking. Many people get stuck with the current system, and it is difficult for them to ideate the future (cf. the relative lack of responses to the question about visions in Paper II). Developers should try to identify such persons among the users with whom collaboration is fertile and with whom it is worthwhile to do development work. Collaboration is easiest when the participants are open-minded and possess knowledge on different professions' tasks. Of course the criteria for selecting participants for usability testing must not be equally tight – it is important to have many kinds of viewpoints present for these tests.

5.2.6 Get a holistic big picture and model the development target to achieve a shared understanding

Lots of factors impact usability in HITS development, including laws and decrees, technical context, etc. (cf. Paper IV). From the perspective of improving usability, it might be helpful to determine which factors have an impact regarding the product being developed. Furthermore, it might be useful to consider all stakeholders with whom it will be necessary to reach a mutual understanding.

There are many different collaborative actions needed to ensure usability. In particular, there should be close tripartite cooperation between national actors, healthcare providers, and software providers.

The agile school thinks that it is difficult to define all requirements in the beginning of the process (Avison & Fitzgerald, 2006). In their opinion, requirements are then collected piece by piece, in an iterative manner. Based on experiences, fundamental requirements must be found out early in order to establish the logic structure of the software in place. The whole must be grasped on a rough level first, and only after that can it be split into pieces that can be refined one iteration at a time.

When a holistic big picture has been achieved, the development target needs to be modelled to achieve a shared understanding. Prototyping and writing goal-based use cases in the beginning of the development phase are proper means to model the development target in a way that every stakeholder understands.

5.2.7 Develop the HITS in close collaboration with users

Users' primary function in development activities should be to act as a source of ideas. Without users it is simply not possible to develop software for such a complex domain as healthcare. It is important to continuously do user / usability testing, usability walkthroughs, and reviews in order to collect feedback, both in the beginning of development by using prototypes (both low and high fidelity) and later on by using already functioning software (e.g., agile sprint releases). Making changes based on the test results is just as important as the testing itself is. The importance of user participation in HITS development has also been highlighted by a number of recent studies (e.g. Høstgaard et al., 2011; B. Rahimi et al., 2014; Shah & Robinson, 2007).

5.2.8 Develop evidence-based design patterns for HITS user interfaces and interaction

Microsoft Health's common user interface (MSCUI) provides User Interface Design Guidance and user interface controls that provide more patient safety to the health software user interface (MSCUI, 2015). In addition to the MSCUI guides and controls, it would be important to create evidence-based user interface models for healthcare software. It would also be fruitful to continue that work and enlarge the set of guidance for user interface and also for interactions. If general standards for healthcare UI could be developed for every provider to follow, life would be easier for users (primarily physicians) who use many software systems in different workplaces, such as public hospitals and private practices. This recommendation confirms earlier research in health informatics (e.g. Armijo et al., 2009).

5.2.9 Inform users about changes and why the changes are made, and provide feedback to users about their 'change requests' and 'development wishes'

Software and information systems change continuously based on laws, regulations, development wishes, or new technology. Today structured data input is mandated by national archive, but it would be beneficial to show and tell users how this method of recording data might help them later on. Also, development wishes may be contradictory among end users – some users are not delighted by the changes that others were awaiting eagerly. It would be challenging but also beneficial to all stakeholders to present the reasoning for the changes that are made.

Users should also be informed about what will happen to the requests or wishes that they have submitted, and when can they expect a reaction or response to their feedback. When the software has been modified due to user feedback, perhaps the sender of the feedback should be acknowledged. That one extra step would be very beneficial for future collaboration, and frustration among end users could decrease.

In HIT development research, however, this kind of collaborative action has not gained much attention.

5.3 MODEL FOR HITS DEVELOPMENT

The recommendations are not sufficient as the only contribution to usability development. Therefore, a model of the HITS development was developed as the ultimate result of this PhD study. It is based on the ADISD model (presented in section 2.2.5) as well as the user-centred design (section 2.3) and agile software development (section 2.2.3) models.

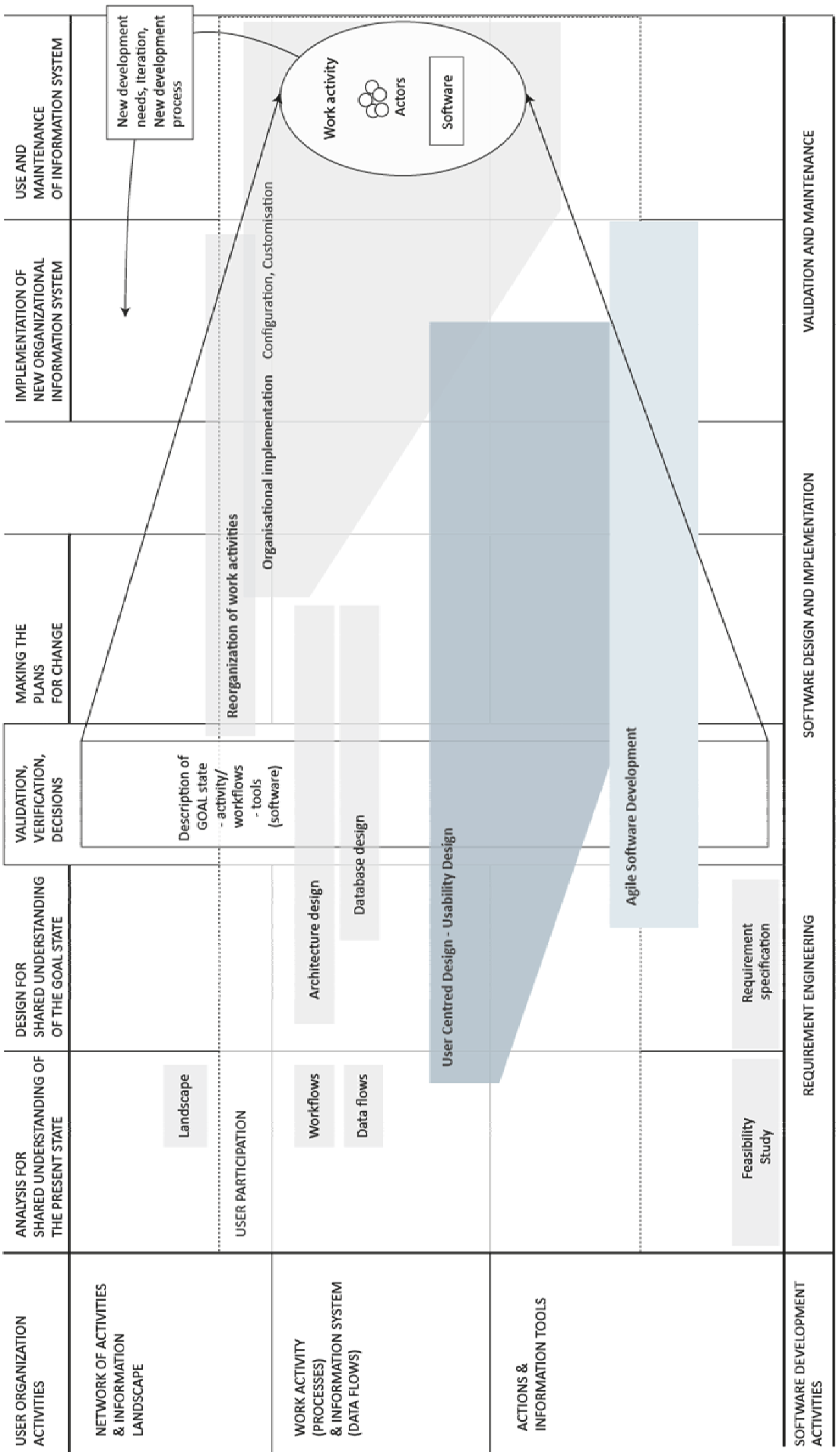


Figure 19. Extended ADISD model - mapping to UCD and agile models

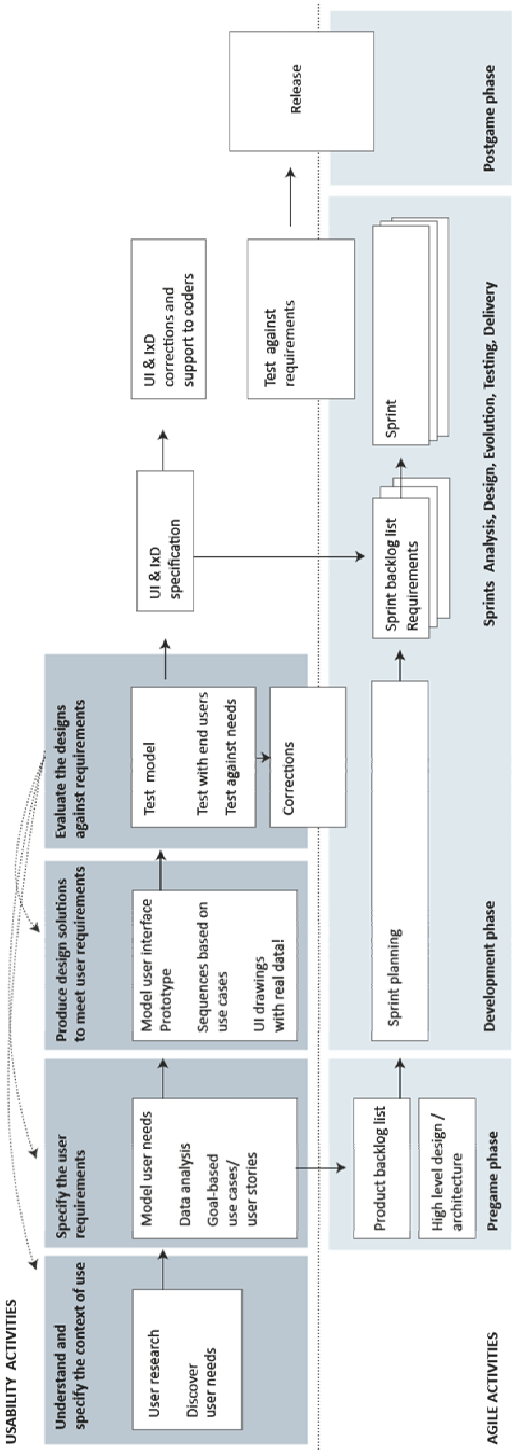


Figure 20. User centred design and agile development, detailed level.

The need to develop work processes and software simultaneously is one of this study's findings (section 5.1.11) and recommendations (section 5.2.1). Since the ADISD model contributes to exactly that issue, it was selected as the starting point (the leftmost three columns at the background of Figure 19). Another reason for using the ADISD model as the starting point was that it could clarify the roles and responsibilities of the persons participating in the development process and allow the participants to see the "big picture" of the development activities. Software development in the healthcare context is demanding, and an ordinary software-focused development model is not enough in this domain. Various actors, activities, and multi-professional activity networks need efficient tools to manage information for patient care; building such tools needs a holistic development model to lean on. The model presented in this chapter has the ambitious goal to be the first draft of such a holistic development model.

The previously existing ADISD model was extended on the basis of sub-studies I-IV and the author's professional experience. In the extended version there are two additional phases that take place in the user organisation: implementing the new organisational information system in it, and the everyday use and maintenance of the information system (the rightmost two columns in Figure 19).

The software development activities identified in Table 2 were then mapped to the ADISD phases (the bottommost row in Figure 19). Naturally, ADISD's level of actions and information tools was the one that most needed further development in order to better fit the model of practical software development. User centred design in the middle of the figure moves from the big picture created in ADISD to the more concrete user-centred design of the software that should be part of the goal state. The user interface (UI), interaction design (IxD), and specifications produced in user-centred design are inputs to the sprint backlog list of the agile software development model (the lowest level in), which was included to bring in the realism from today's software development industry.

The model presented here is a proposal and needs to be tested in real development projects. While testing it, improvements to the model should be documented.

5.3.1 Description of the goal state

It is important to keep in mind the goal state, which in the HITS development case consists of a description of the activities and the software. This description could be part of the requirements specification phase. The activities – healthcare workers' work processes – can be described with work flow charts and goal-based use cases. The most efficient means of describing the software artefact, a tool, is prototyping. The prototype may be low-fidelity (paper prototype) or high-fidelity (functional prototype). This description is created in the first two phases inherited from ADISD ('analysis for shared understanding' and 'design for shared understanding'), and it is validated in the 'validation, verification, decisions' phase. The first two phases and experience of carrying them out has been described in previous research (e.g., Luukkonen et al., 2013; Luukkonen, 2012; M'Rithaa, 2015; Pentikäinen, 2014).

5.3.2 Reorganisation of work activities

According to the model, the next part in the HITS development process consists of the simultaneous development of the work and the software. The first side of it (called reorganisation of work activities in Figure is not described in depth in this thesis, although it is an important factor for the end-users to be satisfied with their software systems. The essential part is to keep developing the work activity continuously while the software is developed, since the former produces much input into the latter. References to various work development approaches and methods are provided, for instance by Pentikäinen (2014).

5.3.3 Constructing the software

Regarding the simultaneous development of work and software, the focus in this thesis is to describe in more detail the software development side of the process. The UCD and agile processes were placed in figure 19 on a general level, while proposals for the combination of these two on a practical, detailed level are presented in Figure 20.

At first, it is necessary to **understand and specify the context of use**. Designers need to discover the healthcare workers', i.e. end-users', needs by visiting their workplaces and by getting familiar with the current tools and processes. This is often called user research. This activity can also be part of two first phases of the ADISD model.

Specifying and modelling user needs and requirements contains two tasks: data analysis and writing goal-based use cases. Data analysis is based on user research and its aim is to find out what information is needed when the user conducts his or her task. A goal-based use case describes the aim and decision points of a specific information processing situation. While writing goal-based use cases, the designer also defines what the new goal is and how to improve the process with the new tools. These use cases are an input for the **product backlog** list. **High-level architecture design** also requires input from the user research and user requirements specification.

It is important to prioritise the user needs to clarify where the focus of planning should be: what is really important to the users and what is not, what must be done first and what can be left for a later day. It is not possible to do everything. For instance, users often wish to have adaptable systems, but it is not clear how many and how widely indeed to modify the system versus just using the default settings.

In the development phase, the designers need to **produce design solutions to meet user requirements** and **evaluate the design against requirements**. This is iterative work. The goal-based use cases are a prerequisite for producing a design solution. When designing software to be used in the healthcare context, the designer needs to produce design solutions with real data. Data gathered in the data analysis phase and applied in the use cases is utilised. Without realistic data in user interface drawings and in prototypes, users focus on the incorrect data and not on the layout or the presentation form of the data, contrary to the aims of the evaluation. At this stage of the design, the focus needs to be on the interaction design, not on a polished user interface design. The cake can be decorated later on, but the interactions are the most important thing.

The outcome of these activities is validated in a **user interface and interaction specification** or a **prototype** that presents all user interfaces for the case. Usability testing methods are utilised in the evaluation. Usability testing for plans and prototypes should be done often enough, and the test users should be the actual end-users or similar to them. Design and validation are iterated until the usability test results are on an acceptable level.

The specification is input to the sprint backlog. During the sprint, designers provide developers with **usability design support to the user interface implementation**. Software testing also takes place in this development phase. Software test cases could be based on the goal-based use cases, but the most important thing is that the test cases correspond to the reality of the end-users.

In the **postgame phase**, the outcome of the construction is validated, be it a stand-alone software system or a piece of software that will be integrated with previously constructed software. Finally, system and acceptability testing are carried out.

5.4 REFLECTIONS ON THE RESULTS

A number of studies have shown usability flaws in healthcare IT systems (e.g., Kaipio, 2011; Kushniruk et al., 2014; Nielsen, 2005; Smelcer et al., 2009), and some studies have proposed or used UCD methods to solve such flaws (e.g Kaufman et al., 2003; Kushniruk & Borycki, 2006; Kushniruk & Patel, 2004; Walji et al., 2013; Yen & Bakken, 2012). This thesis extends the perspective further by arguing that using UCD methods in HITS development is not a sufficient solution to usability problems, and a bigger view must be achieved to develop high usability HITS. A few subjective reflections on health informatics, based on the analysis of the findings, are presented next.

The people who participate in HITS development and their skills are very significant. Therefore, such usability design and requirement analysis experts who specialise in healthcare are needed. People who work with users, such as requirements analysts (in software provider companies, in healthcare organisations, and in national authority organisations), testers, concept designers, and clinical experts might benefit from specialised training. In Finland, physicians have the possibility to study Healthcare Informatics Specialist qualifications (in Finnish: Terveydenhuollon tietotekniikan erityispätevyys) (FMA, 2015). Should that kind of specialised education be required of all stakeholders in HITS development?

One of the recommendations in this thesis is "Plan user participation and carefully choose the users who participate". In addition to that, software provider organisations should plan which persons from their organisations communicate with users and how the information from the users should be spread to all who need it in their own work, to avoid interaction breakdowns. Healthcare organisations should also carefully consider which persons they send to present users in HITS development activities.

New technological possibilities provide new means to usability and user participation in HITS development. Automatic user participation, i.e. using big data to understand the interactions between users and the computer, is presented by Kushniruk et al. (2014 p. 80): "*… massive interaction data, 'big data', will require new types of methods to be used in the gathering and analysis of user data for remotely collecting and analyzing user interactions with online clinical information resources (e.g. clinical guidelines) and the increasingly varied range of health information systems being deployed online*".

End-user organisations must **demand** usability. To achieve high usability, it is not enough to have a requirement that "the system should be easy to use" (Jokela, 2011). Usability requirements should be based on and formulated with in-depth understanding of users' workflows. To be re-emphasised as the last word, user work and software need to be developed simultaneously, as proposed in section 5.2.

6 Discussion

Conducting this PhD thesis research - learning new and revising already familiar topics – was an extremely exiting journey for the author. The study as a whole is assessed in this discussion chapter. The answers to the research questions are assessed first. Then the contributions of the thesis to research and practice are discussed, followed by a consideration of the strengths and limitations. Finally, ideas for future research work are presented.

6.1 ANSWERS TO THE RESEARCH QUESTIONS

The answers to the sub-research questions are assessed first; then the answer to the overall research question is presented.

SRQ1: Are user-centred methods useful in HITS development?

In sub-study 1 (paper I), preliminary experiments concerning the use of UCD methods in healthcare IT system development were conducted. The results were extremely positive: end-users were enthusiastic to participate in user-centred activities provided by the researchers: they allowed the researchers to conduct a heuristic analysis on their current solution, participated in rating the findings, involved the researchers in a ward round, actively commented on user interface drawings (i.e., low fidelity prototypes), and participated in usability walkthroughs. In the feedback questionnaire, every respondent regarded user interface drawings as were extremely useful when user needs were collected.

Literature supports the findings that UCD methods are useful in the healthcare context (Butz & Krüger, 2007; Pohl et al., 2011; N. Rahimi & Ibarra, 2014; Urda et al., 2013). A recent systematic review of UCD practices in healthcare (Ghazali, Nurul, & Omar, 2014) concludes that UCD can also indirectly help to provide "*intended health outcomes that can please the end users whether the patients or the medical practitioner"*. An answer to SRQ1 was thus achieved, but the use of UCD methods should be continuously studied to keep the methods up to date.

SRQ2: What is the current state of the usability of HITS and user participation in HITS development from the user viewpoint?

SRQ3: What is the current state of the usability of HITS and user participation in HITS development from the developer viewpoint?

Sub-research questions 2 and 3 are dealt with together as a pair here, because the answers to them form a sensible and coherent whole.

The physicians who responded to the questionnaire study presented in Paper II had quite negative experiences of both the usability of the information technology systems they used and the methods of participation. The study showed that physicians were willing to participate in some ways in the development of HITS if the conditions to participate and the means were fair and correct in their point of view. Sub-study II stressed that user participation and the software development methods need to be developed further to better fit the complex healthcare IT systems development context. Physicians also claimed that the usability of the current HITS is poor, but they presented only few visions of future HITS.

In sub-study 3, HITS developers had quite positive opinions of the present state of the usability of HITS and the user participation in HITS development. In spite of the positive results, further development of the methods is recognised. Paper III presents how the end-users' and developers' opinions and views differ, most strikingly about whether developers are indeed interested in end-users' views and whether developers really take end-users' opinions into account. Both groups agreed that the development pace of HITS was too slow.

Agile development methods are often offered as a solution to the slowness of development, because agile methods are argued to result in faster software development (Abrahamsson et al., 2002). Research literature also stresses using user-centred methods to take end-user needs and views better into consideration in the software development process.

Knowledge of the current state of the usability of HITS and the user participation in HITS development as seen from both ends was achieved with these two questionnaire studies. Results from sub-study 4 reinforced the results of sub-studies 2 and 3.

SRQ4: Which issues in addition to user participation affect the usability development of HITS?

A landscape view of HITS development was depicted in Paper IV. A significant number of stakeholders influence HITS usability development. In order to develop high usability software, there is a need to understand how all identified stakeholders affect the development target. The stakeholder map, created on the basis of expert interviews, offers a good starting point to achieve a fuller understanding of the big picture. A number of issues that affect the usability development of HITS were found through the interviews conducted in sub-study 4.

SRQ5: What can be done to improve the usability of HITS and user participation in HITS development?

Knowing developers' and end-users' views on HITS development made it possible to suggest means and methods for HITS development activities. The findings from all sub-studies were combined in section 5.1. The findings form the basis for recommendations on how usability could be improved. After analysing all the materials from all sub-studies, proposals for usability development were formed and presented in section 5.2. In total, 9 recommendations (sub-sections 5.2.1 - 5.2.9) were formed. Furthermore, the activity-driven information system development model ADISD (Toivanen et al., 2007; 2009) was extended by combining both agile and UCD development models with it (section 5.3). The combined model may act as a basis for detailed software process improvements in HITS provider companies.

In conclusion, six recommendations were formed to improve the **usability of HITS**:

- Develop the end-users' work practice and the software simultaneously.
- Increase developers' domain knowledge.
- Identify stakeholders that impact and have responsibility to design the good usability of software.
- Make the activities of the software development process more visible to different stakeholders.
- Get a holistic big picture and model the development target to achieve a shared understanding.
- Develop evidence-based design patterns for HITS user interfaces and interaction.

Three recommendations were formed to improve **user participation in HITS development**:
- Plan user participation and carefully choose the users who participate.
- Develop the HITS in close collaboration with users.
- Inform users about changes and why the changes are made, and provide feedback to users about their 'change requests' and 'development wishes'.

Lastly, the response to the overall research question is presented.

RQ: What factors impact the usability of HITS and user participation in HITS development in Finland?

It can be summarised that the following factors impact HITS development: software development and work development methods and means (including user-developer collaboration); national and international legislation and regulation; the views and attitudes of developers and end-users themselves; and also management work on both sides. The depth and intensity of the collaboration between participants, in particular tripartite cooperation (between national actors, software providers, and healthcare providers) is one of the key factors.

A number of factors affect both the usability of HITS and the end-users' experience of using IT systems to support their daily work. Usability / UCD methods are extremely important, but they alone cannot heal healthcare IT systems' usability issues. Every participant should familiarise himself or herself with the big picture of healthcare IT development to see how one's own piece of work affects the whole.

6.2 CONTRIBUTIONS OF THE THESIS

Research can make contributions in terms of providing new scientifically acquired empirical knowledge of a phenomenon (knowledge contribution), new theoretical understanding of a phenomenon (theoretical contribution), new ways of conducting research (methodological contribution), or results that can be applied in practice (practical relevance).

Papers II-IV produced previously non-existent empirical knowledge concerning the current state of and development spots in the usability of HITS and HITS development in Finland. Methodologically, this research aimed at applying existing ways of doing research rather than developing new ones. The way of constructing the big picture presented in Paper IV can, however, be regarded as a new research-methodological contribution. The theoretical and practical contributions are assessed separately in the following sub-sections.

6.2.1 Theoretical contribution

In the end of literature review, it was found that the gap in knowledge where this thesis aims to contribute is 'why usability of healthcare IT systems is poor, even if solutions for improvement are theoretically known'.

What new does this study bring to theoretical knowledge?

The main result of this thesis is to combine the analysis and concrete development activities in a new way, extending the ADISD model with UCD and agile methods. The ADISD model has been tried in the early phases of information systems and software development with positive results, as mentioned in section 2.2.5. The ADISD approach brings about a more holistic view of software development than models like agile methods, which focus on the software development process only. In the complex domain of healthcare, a holistic development model is needed to achieve high usability tools for healthcare workers.

This study also confirms earlier findings of the critical importance of user participation in HITS development (e.g. Høstgaard et al., 2011; Kaipio, 2011). As a step forward, in this study user participation is linked to a holistic model for HITS development. Stakeholders' holistic understanding of the development of HITS was noticed to be an important factor. This is in line with previous research (Dittrich & Heje, 2006) that stresses the importance of understanding each other's work when developing information systems and software systems.

The need for the big picture in the beginning of the development has been known for a while. For instance, Hansson et al. (2006) stressed the need for a holistic view when combining agile-like and participatory design methods in software development. However, the kind of map that was constructed in sub-study 4 was not found while conducting the literature review.

Usability and user participation issues have been discussed and researched over years in the areas of software development, information system development, human computer interaction usability, and user-centred design. Healthcare brings its own challenges to usability and user participation in software development, and that is the issue that has gained less research attention.

This study makes a contribution to the discipline of health informatics; empirical knowledge about the problems in practice was obtained, and based on that, proposals were presented. Although this subject has been studied previously in the discipline of the health informatics, this research could have a wider contribution to software development, usability, and user participation.

6.2.2 Practical relevance

In sub-study 3, it was concluded that the key issue is no longer to raise awareness of the importance of user-centred design and end-user participation in HITS provider companies. Despite all the recommendations in the literature on user-centred design, agile methodology, or other related methods, it is still not clear how the good intentions should be implemented in the practice of HITS product development. Many models are presented for software development in academic research, but few practical experiences on those models are studied. This thesis suggests practical ways to improve the usability of HITS by suggesting recommendations and a development model. The results thus have direct practical relevance to HITS provider companies and HITS user organisations, at least in Finland and arguably more widely, as well.

6.3 STRENGTHS AND LIMITATIONS OF THE STUDY AND FUTURE RESEARCH

In the researcher's own assessment, the **strengths** of the study include, basically, that it forms a holistic entity despite being a set of individual research articles; its scope is clearly defined, its size fits the requirements for a PhD study, and clear answers to the research questions were achieved. The research papers have a strictly planned structure and, together with this summary section, they form a coherent continuum. That the thesis provides both straightforward recommendations and the enriched model can be considered another strength. The exceptionally large and representative sample in Paper II is a strength, as well. The research is reported in such a way that it can be repeated in other countries.

The main **limitation** in this study is that the recommendations and model could not be tested in practice within the scope of a single PhD. From an international perspective, it is a limitation that the empirical research was conducted in only one country, and the results thus cannot be generalised to other countries without caution. That the main focus was on only one user group, the physicians, in two of the sub-studies (Papers II and III) can be considered another limitation. As pointed out in Paper I, physicians have been a very critical user group of HITS, and they have expressed their opinions in public, which was a reason for studying their opinions and experiences more deeply. In the last paper (IV), the perspective was wider and other professionals from different perspectives were involved. In Paper III, the sample consisted of only one HITS provider company, although it is one of the major ones. As discussed in Paper III, the results from small companies could be different.

Further research should address the limitations of the current study. Research should be conducted on how the recommendations and the development process model work in real development contexts, and how they should be further developed. It would also be interesting to know more about other countries' HITS development landscapes – what are the differences and the similarities? It would be interesting to know about other health professionals' views, as well, on HITS usability, development activities, and participation. It would be extremely interesting to develop interaction design and user interface design patters for HITS. Beyond research, there is a need to develop training and education for the usability and requirement analysis experts working in the healthcare context.

7 Conclusion

Both international and Finnish national studies have shown that the usability and user experiences of healthcare IT systems have not been at a sufficiently high level, from the end-users' point of view. As Winter (2013) states, *"Achieving good usability and UX is hard."* It is a difficult task, but not unreachable. The methods and means used in the development of HITS need to be revised and developed further in order to achieve good usability.

The aim of this PhD research was accordingly to study what factors impact the usability of HITS as well as the user participation in HITS development in Finland, with the objective of finding ways of improving the current situation.

User-centred design methods were tested, developed, and shown to be useful in an exploratory way in the context of healthcare IT development.

User participation is known to be one of the main success factors in information systems and software systems development. In this study, the current state of end-user participation in the HITS development scene in Finland was investigated. Both end-users' (in this case physicians') and developers' views were studied, in 2010 and 2013 respectively. User participation was found not to be the only issue affecting the usability of HITS. Studying and modelling the landscape view of HITS development revealed that a number of stakeholders and factors caused challenges for usability development.

Recommendations towards building better usability were formed on the basis of the research findings. The recommendations emphasise two things in particular: every stakeholder's understanding of the big picture of both software development activities and healthcare workers' activities, and carefully planned user participation throughout the development of healthcare information systems (including IT products and work activities) in all organisations involved in the development.

In addition to the practicable recommendations, the activity-driven information system development model (ADISD) was further developed to better fit the industry practice, and therefore extended with two new phases and combined with UCD and agile models. By the recommendations and the enriched model, this thesis thus contributes to the objective of progressing towards better usability of healthcare IT systems.

Future research is needed to test the proposed model in real-life development projects and to compare the results from Finland with the situation in other countries.

References

Aaen, I., Arent, J., Mathiassen, L., & Ngwenyama, O. (2001). A Conceptual MAP of Software Process Improvement. *Scandinavian Journal of Information Systems*.

Abrahamsson P. (2002). *The Role of Commitment in Software Process Improvement*. University of Oulu.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods*.

Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). DRAFT: User-Centered Design. In *W. Encyclopedia of Human-Computer Interaction*. doi:10.3233/WOR-2010-1109

Adler, P. S. (2005). The Evolving Object of Software Development. *Organization*, *12*(3), 401–435. doi:10.1177/1350508405051277

Alharthi, H., Youssef, A., Radwan, S., Al-Muallim, S., & Zainab, A.-T. (2014). Physician satisfaction with electronic medical records in a major Saudi Government hospital. *Journal of Taibah University Medical Sciences*, *9*(3), 6–11. doi:10.1016/j.jtumed.2014.01.004

Armijo, D., McDonnell, C., & Werner, K. (2009). Electronic Health Record Usability: Interface Design Considerations. Rockville, MD: Agency for Healthcare Research and Quality.

Avison, D., & Fitzgerald, G. (2006). *Information systems development methodologies, technologies & tools* (4th Editio.). Berkshire: McGraw-Hill Companies.

Barki, H., & Hartwick, J. (1989). Rethinking the concept of user involvement. *MIS Quarterly*, *13*, 53–63. doi:10.2307/248700

Barki, H., & Hartwick, J. (1994). Measuring User Participation, User Invovlement, and User Attitude. *MIS Quarterly*, *18*(1), 59–82. doi:10.2307/249610

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of AIS*, *2*, 4. doi:http://www.cis.gsu.edu/~rbaskerv/CAIS_2_19/CAIS_2_19.html

Beck, K., Beedle, M., van Bennekum, A. Cockburn, A. Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., … Thomas, D. (2001). The agile manifesto. *The agile alliance.* Retrieved July 24, 2013, from http://agilemanifesto.org

Belden, J. L., Grayson, R., & Barnes, J. (2009). Defining and Testing EMR Usability : Principles and Proposed Methods of EMR Usability Evaluation and Rating HIMSS.

Bevan, N. (1997). Quality in use: incorporating human factors into the software engineering lifecycle. *Proceedings of IEEE International Symposium on Software Engineering Standards*, *June*, 169–17. doi:10.1109/SESS.1997.595963

Bevan, N. (2008). UX , Usability and ISO Standards. In *CHI 2008* (pp. 1–5).

Beyer, H. (2010). User-Centered Agile Methods. Morgan&Claypoll Publishers.

Beyer, H., & Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centered System*. San Francisco: Morgan Kaufmann Publishers.

Bhattacherjee, A. (2012). *Social science research: principles, methods, and practices* (2nd editio.). Retrieved from http://scholarcommons.usf.edu/cgi/viewcontent.cgi?article=1002&context=oa_textbooks

Boehm, B. (1996). Anchoring the Software Process. *Software, IEEE*, 73–82.

Boivie, I., Gulliksen, J., & Göransson, B. (2006). The lonesome cowboy: A study of the usability designer role in systems development. *Interacting with Computers*, *18*(4), 601–634. doi:10.1016/j.intcom.2005.10.003

Bosch, J. (2009). From Software Product Lines to Software Ecosystems. *Proceedings of the 13th International Software Product Line Conference*, (Splc), 111–119. doi:10.1016/j.jss.2012.03.039

Bosch, J., & Bosch-Sijtsema, P. (2010). From integration to composition: On the impact of software product lines, global development and ecosystems. *Journal of Systems and Software*, *83*(1), 67–76. doi:10.1016/j.jss.2009.06.051

Bostrom, R. P. (1989). Successful application of communication techniques to improve. *Information & Management*, *16*(5), 279–295. doi:10.1016/0378-7206(89)90005-0

Brown, T. (2008). Design thinking. doi:10.1037/e447202008-010

Butz, A., & Krüger, A. (2007). User-centered development of a pervasive healthcare application. *2006 Pervasive Health Conference and Workshops, PervasiveHealth*. doi:10.1109/PCTHEALTH.2006.361650

Bødker, S., & Greenbaum, J. (1993). Design of Information Systems : Things versus People. In *Things versus people. Green et al (eds.) Gendered by Design, 53-63.* (pp. 53–63).

Cavaye, A. L. M. (1995). User participation in system development revisited. *Information & Management*, *28*(5), 311–323. doi:10.1016/0378-7206(94)00053-L

Chamberlain, S., Sharp, H., & Maiden, N. (2006). Towards a Framework for Integrating Agile Development and User-Centred Design. *Extreme Programming and Agile Processes in Software Engineering*, *4044/2006*, 143 – 153. doi:10.1007/11774129_15

Chan, J., Shojania, K. G., Easty, A. C., & Etchells, E. E. (2011). Does user-centred design affect the efficiency, usability and safety of CPOE order sets? *Journal of the American Medical Informatics Association : JAMIA*, *18*(3), 276–281. doi:10.1136/amiajnl-2010-000026

Christensen, T., Faxvaag, A., Loerum, H., & Grimsmo, A. (2009). Norwegians GPs' use of electronic patient record systems. *International Journal of Medical Informatics*, *78*(12), 808–14. doi:10.1016/j.ijmedinf.2009.08.004

Christensson, P. (2015). The Tech Terms Computer Dictionary. Retrieved February 14, 2015, from http://techterms.com/definition/user_interface.

Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3 The Essentials of Interaction Design*. Indiana: Wiley Publishing, Inc.

Cooper, A., & Saffo, P. (1999). *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How To Restore The Sanity*. Sams.

CSIAC. (2013). *Website of The Cyber Security and Information Systems Information Analysis Center*. Retrieved July 26, 2013, from https://sw.thecsiac.com/databases/ url/key/39

Da Silva, T. S., Martin, A., Maurer, F., & Silveira, M. (2011). User-centered design and agile methods: A systematic review. *Proceedings - 2011 Agile Conference, Agile 2011*, 77–86. doi:10.1109/AGILE.2011.24

Da Silva, T. S., Silveira, F. F., Silveira, M. S., Hellmann, T., & Maurer, F. (2015). A Systematic Mapping on Agile UCD Across the Major Agile and HCI Conferences. In O. et Al Gervasi (Ed.), *ICCSA 2015, Part V, LNCS 9159,* (pp. 86–100). doi:10.1007/978-3-319-21413-9

DAISY. (2015). Research and Development of human Activity and Information SYstems. *Uef School of Computing Research groups*. Retrieved August 4, 2015, from http://www2.uef.fi/fi/act/research

Dearden, A., Wright, P., Bowen, S., Cobb, M., & Wolstenholme, D. (2010). User centred design and pervasive health : a position statement from the user centred healthcare design project, 8–11. Retrieved from http://pervasivehealth.org/2010/

Dittrich, Y. (2014). Software engineering beyond the project – Sustaining software ecosystems. *Information and Software Technology*, *56*(11), 1436–1456. doi:10.1016/j.infsof.2014.02.012

Dittrich, Y., & Heje, L. P. (2006). ERP Systems meet Participatory Design Socio-Technical Design as a Common Issue ? In *IRIS 29, Helsingør, Denmark, August 12-15, 2006.* (pp. 1–16).

Eastwood, B. (2013). 12 Ways to Improve the Healthcare User Experience. Retrieved October 26, 2014, from http://www.cio.com/article/2386948/healthcare/12-ways-to-improve-the-healthcare-user-experience.html

EKSIG. (2015). Experiential Knowledge Special Interest Group Website. Retrieved August 15, 2015, from http://www.experientialknowledge.org/

Eronen, J. (2007). Tietokoneet vaarantavat potilasturvallisuuden. Retrieved from http://blogit.mtv3.fi/kolmevarttia/2007/08/

Fægri, T. E. (2011). *Collaborative learning in software development : An investigation of characteristics , prerequisites and improvement*. Norwegian University of Science and Technology.

Ferreira, A., Cruz-Correia, R., Chadwick, D., & Antunes, L. (2008). Improving the implementation of access control in EMR. *Proceedings - International Carnahan Conference on Security Technology*, 47–50. doi:10.1109/CCST.2008.4751275

Ferreira, a., Cruz-Correia, R., & Antunes, L. (2011). Usability of authentication and access control: A case study in healthcare. *Proceedings - International Carnahan Conference on Security Technology*. doi:10.1109/CCST.2011.6095873

FMA. (2015). Terveydenhuollon tietotekniikka. *The Finnish Medical Association official website*. Retrieved August 16, 2015, from https://www.laakariliitto.fi/koulutus/erityispatevyydet/tietotekniikka/

Fox, B. D. (2010). *Agile Methods and User-Centered Design: How These Two Methodologies are Being Integrated in Industry*. UNIVERSITY OF CALGARY Agile.

Fox, D., Sillito, J., & Maurer, F. (2008). Agile methods and user-centered design: How these two methodologies are being successfully integrated in industrys. *Proceedings - Agile 2008 Conference*, 63–72. doi:10.1109/Agile.2008.78

García-Mireles, G. A., Moraga, M. A., Garcia, F., & Piattini, M. (2013). The Influence of Process Quality on Product Usability : A Systematic Review. *CLEI Electronic Journal*, *16*(2), 1–13.

Ge, X., Paige, R. F., & McDermid, J. a. (2009). Domain analysis on an electronic health records system. *Proceedings of the First International Workshop on Feature-Oriented Software Development - FOSD '09*, 49. doi:10.1145/1629716.1629727

Ghazali, M., Nurul, A. M. A., & Omar, R. (2014). User Centered Design Practices in Healthcare : A Systematic Review. In *2014 3rd International Conference on User Science and Engineering (i-USEr) User* (pp. 91–96).

Goldkuhl, G. (2004). Meanings of Pragmatism : Ways to conduct information systems research. *In Procedings of 2nd Conf on Action in Language, Organisations and Information Systems(ALOIS2004)*, 13–26.

Greenbaum, J., & Kyng, M. (1991). *Design at work: Cooperative design of computer systems.* Hillsdale NJ.: Erlbaum.

Haikala, I., & Märijärvi, J. (2002). *Ohjelmistotuotanto* (7. renewed.). Vantaa: Talemtum Media Oy.

Halvorson, K. (2010). Intentional Communication : Expanding our Definition of User Experience Design. *Interactions*, *17*(3), 75–77.

Hanssen, G. K. (2012). A Longitudinal Case Study of an Emerging Software Ecosystem: Implications for Practice and Theory. *Journal of Systems and Software*, *85*(7), 1455–1466. doi:10.1016/j.jss.2011.04.020

Hansson, C., Dittrich, Y., & Randall, D. (2006). How to Include Users in the Development of Off-the-Shelf Software: A Case for Comple- menting Participatory Design with Agile Development. In *Proceedings of the 39th Hawaii International Conference on System Sciences*.

Harris, M. A., & Weistroffer, H. R. (2008). Does User Participation Lead to System Success? *Proceedings of the Southern Association for Information Systems Conference*, 1–6. Retrieved from http://sais.aisnet.org/2008/1BHarrisWeistroffer.pdf

Harrison, M. I., Koppel, R., & Bar-Lev, S. (2007). Unintended consequences of information technologies in health care—an interactive sociotechnical analysis. *Journal of the American Medical Informatics Association, 14(5)*, 542–549.

Hassenzahl, M., & Tractinsky, N. (2006). User experience - a research agenda. *Behaviour & Information Technology*, *25*(2), 91–97. doi:10.1080/01449290500330331

Heeks, R. (2006). *Health information systems: Failure, success and improvisation*. *International Journal of Medical Informatics* (Vol. 75). doi:10.1016/j.ijmedinf.2005.07.024

HIMSS. (2015). Medical Informatics. Retrieved November 4, 2015, from http://www.himss.org/clinical-informatics/medical-informatics

Hix, & Hartson. (1993). *Developing User Interfaces: Ensuring Usability Through Product & Process*. John Wiley & Sons.

Hodgson, P. (2010). Usability for Medical Devices A New International Standard ISO IEC 62366. *UserFocus website*. Retrieved from http://www.userfocus.co.uk/articles/ISO62366.html

Holmlid, S. (2005). Service Design methods and UCD practice. *Proceedings of the INTERACT 05-Workshop*, 63–68. doi:10.3109/11038128.2011.556196

Humayoun, S. R., Dubinsky, Y., & Catarci, T. (2009). A Three-Fold Integration Framework to Incorporate User–Centered Design into Agile Software Development. In *HCI (16)* (pp. 55–64). doi:10.1007/978-3-642-02806-9

Hwang, M. I., & Thorn, R. G. (1999). The effect of user engagement on system success: A meta-analytical integration of research findings. *Information & Management*, *35*(4), 229–236. doi:10.1016/S0378-7206(98)00092-5

Hyysalo, S. (2006). *Käyttäjätieto ja käyttäjätutkimuksen menetelmät*. Helsinki: Edita Prima Oy.

Hyysalo, S. (2010). *Health Technology Development and use From Practice-Bound Imagination to Evolving Impacts*. Routledge, New York.

Høstgaard, A., Bertelsen, P., & Nøhr, C. (2011). Methods to identify, study and understand End-user participation in HIT development. *BMC Medical Informatics and Decision Making*, *11*(1), 57. doi:10.1186/1472-6947-11-57

International Standard ISO / IEC 2007. (2008). INTERNATIONAL STANDARD ISO / IEC 12207 IEEE Std 12207-2008.

ISO. (2009). ISO 9241-210 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en

Ives, B., & Olson, M. H. (1984). User involvement and MIS success: A Review of research. *Management Science*, *30*(5).

Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Administrative Science Quarterly*, *24*(4), 602. doi:10.2307/2392366

Johannessen, L. K., & Ellingsen, G. (2009). Integration and Generification—Agile Software Development in the Healthcare Market. *Computer Supported Cooperative Work (CSCW)*, *18*(5-6), 607–634. doi:10.1007/s10606-009-9097-8

Johnson, C. ., Johnson, T. D., & Zhang, J. (2005). A User-centered Framework for Redesigning Health Care Interfaces. *Journal of Biomedical Informatics*, *38*(1), 75–87.

Johnson, C. M., Johnston, D., Crowley, P. K., Bulbertson, H., Rippen, H. E., Damico, D. J., & Plaisant, C. (2011). EHR Usability Toolkit: A Background Report on Usability and Electronic Health Records. Rockville, MD.: AHRQ Publication No. 11-0084-EF.

Jokela, T. (2008). Tietojärjestelmästä voi saada vaivattoman. *Suomen lääkärilehti2*, *34*.

Jokela, T. (2010). ISO 9241-210 Human-centred design for interactive systems . Mitä se on ? *Blogi: ISO 9241-210 Human-centred design for interactive systems. Mitä se on?* Retrieved July 9, 2015, from http://iso9241-210.blogspot.fi/2010/04/iso-9241-210-on-ilmestynyt-korvaa-iso.html

Jokela, T. (2011). Terveydenhuollon tietojärjestelmät - sitä saa miten tilaa. *Finnanest*, *44*, 219–222.

Jokela, T. (2015). Video 2: Mitä jokaisen hankintaan osallistuvan tulisi tietää käytettävyydestä. Mitä käytettävyys on - ja mitä se ei ole. Entä mitä on käyttäjäkokemus? *Web course: Käytettävyys tietojärjestelmien hankinnoissa.* Retrieved April 20, 2015, from https://tjokela.leadpages.net/l-kth-video-2/

Jokela, T., & Abrahamsson, P. (2004). Usability Assessment of an Extreme Programming Project: Close Co-operation with the Customer Does Not Equal to Good Usability. In *Product Focused Software Process Improvement* (pp. 393–407). doi:10.1007/b96726

Jokela, T., Iivari, N., Matero, J., & Karukka, M. (2003). The Standard of User-Centered Design and the Standard Definition of Usability : Analyzing ISO 13407 against ISO. *Design*, *46*, 53–60. doi:10.1145/944519.944525

Jones, P. H. (2013). *Design for Care: Innovating Healthcare Experience*. New York: Rosenfeld Media.

JYU. (2015). Pragmatismi. *Koppa. University of Jyväskylä*. Retrieved November 8, 2015, from https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tieteenfilosofiset-suuntaukset/pragmatismi

Kaipio, J. (2011). *Usability in Healthcare: Overcoming the Mismatch between Information Systems and Clinical Work.* Doctoral dissertation. Department of Computer Science and Engineering, Aalto University.

Kaufman, D. R., Patel, V. L., Hilliman, C., Morin, P. C., Pevzner, J., Weinstock, R. S., … Starren, J. (2003). Usability in the real world: assessing medical information technologies in patients' homes. *Journal of Biomedical Informatics*, *36*(1-2), 45–60. doi:10.1016/S1532-0464(03)00056-X

Keil, M., & Carmel, E. (1995). Customer-developer links in software development. *Communications of the ACM*, *38*(5), 33–44. doi:10.1145/203356.203363

Keinonen, T. (2009). Design Contribution Square. *Advanced Engineering Informatics*, *23*(2), 142–148. doi:10.1016/j.aei.2008.10.002

Korhonen, M. J., & Hartikainen, S. (2009). Löytyykö potilaan lääkitys sähköisestä sairauskertomuksesta? *Suomen Lääkärilehti*, *64*(1-2), 60–62.

Korpela, M., Soriyan, H. A., & Olufokunbi, K. C. (2000). Activity analysis as a method for information systems development: General Introduction and Experiments from Nigeria and Finland. *Scandinavian Journal of Information Systems*, *12*(1).

Korpimies, A. (2015). It-järjestelmä tuhlaa lääkärin työaikaa. *TIVI Tiedon Ja Tekniikan Ytimessä*, (January), 10.

Kujala, S. (2002). *User Studies: A Practical Approach to User Involvement for Gathering User Needs and Requirements*. Espoo: Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 116, 132 pp.the Finnish Academies of Technology.

Kujala, S. (2008). Effective user involvement in product development by improving the analysis of user needs. *Behaviour & Information Technology*, *27*, 457–473. doi:10.1080/01449290601111051

Kushniruk, A., Kaipio, J., Nieminen, M., Hyppönen, H., Lääveri, T., & Nohr, C. (2014). *Human Factors in the Large : Experiences from Denmark , Finland and Canada in Moving Towards Regional and National Evaluations of Health Information System Usability Contribution of the IMIA Human Factors Working Group 3 Regional Comparison of the Usabili*. *IMIA Yearbook of Medical Informatics*.

Kushniruk, A. W., & Borycki, E. M. (2006). Low-cost rapid usability engineering: designing and customizing usable healthcare information systems. *Healthcare Quarterly (Toronto, Ont.)*, *9*(4), 98–100, 102.

Kushniruk, A. W., & Patel, V. L. (2004). Cognitive and usability engineering methods for the evaluation of clinical information systems. *Journal of Biomedical Informatics*, *37*, 56–76. doi:10.1016/j.jbi.2004.01.003

Kuusinen, K., Mikkonen, T., & Pakarinen, S. (2012). Agile User Experience Development in a Large Software Organization : Good Expertise but Limited Impact. *Human-Centered Software Engineering. Springer Berlin Heidelberg.*, 94–111.

Laakso, S., & Laakso, K.-P. (2004). Ensuring quality of the user interface with the GUIDe project model. unpublished paper (October 3, 2004). Retrieved from www.cs.helsinki.fi/u/salaakso/papers/GUIDe.pdf

Laakso, S., & Laakso, K.-P. (2015). GUIDe process model. *Interacta Design Oy official web site*. Retrieved July 27, 2015, from http://www.interacta.fi/guide_en.html

Law, E., & Roto, V. (2009). Understanding, scoping and defining user experience: a survey approach (pp. 719–728). Retrieved from http://dl.acm.org/citation.cfm?id=1518813

LeRouge, C., Ma, J., Sneha, S., & Tolle, K. (2013). User profiles and personas in the design and development of consumer health technologies. *International Journal of Medical Informatics*, *82*(11), e251–e268.

Lilja, K. K. (2013). *Differences in Organizational Cultures – A Challenge for IT Projects*. Tampere University of Technology.

Lin, W. T., & Shao, B. B. M. (2000). The relationship between user participation and system success: a simultaneous contingency approach. *Information & Management*, *37*(6), 283–295. doi:10.1016/S0378-7206(99)00055-5

Luukkonen, I. (2012). *Activity-Driven Needs Analysis and Modeling in*. University of Eastern Finland.

Luukkonen, I., Toivanen, M., Mursu, A., Saranto, K., & Korpela, M. (2013). Researching an Activity-Driven Approach to Information Systems Development. In *Miranda, I.M., Cruz-Cunha, M.M., and Gonçalves, P. (eds.) Handbook of Research on ICTs and Management Systems for Improving Efficiency in Healthcare and Social Care* (pp. 431 – 450). IGI Global. Retrieved from http://www.uef.fi/fi/act/publications

Lynch, T., & Gregor, S. (2004). User participation in decision support systems development: influencing system outcomes. *European Journal of Infomation Systems*, *13*(4), 286–301.

Lääveri, T. (2008a). Lääkäriliiton eHealth valiokunta. Potilaskertomusjärjestelmien arvioinnit. *Suomen Lääkärilehti*, *63*(6). Retrieved from http://www.laakarilehti.fi/files/nostot/nosto6_1.pdf

Lääveri, T. (2008b). Mitä kliinikko odottaa sähköisten potilaskertomusten käytettävyydeltä? *Nuori Lääkäri*, (7).

Lääveri, T. (2008c). Potilaskertomusjärjestelmien arvioinnit [Evaluations of patient record systems. In Finnish]. *Finnish Medical Journal*, (6), 8–31.

Lääveri, T. (2008d). Potilaskertomusjärjestelmien käytettävyysselvitys osoitti: Lääkärien mielipiteitä ei ole kuunneltu. *Suomen Lääkärilehti, Potilastietojärjestelmät –liite*, *63*(6), 3.

M'Rithaa, D. K. . (2015). *A Framework for Information Communication that Contributes to the Improved Management of the Intrapartum Period*. Cape Peninsula University of Technology: Cape Town, South Africa.

Mager, B. (2009). Service design as an emerging field. In S. Miettinen & M. Koivisto (Eds.), *Service Design with innovative methods* (pp. 28–43). Keuruu: Otava book printings LTD.

Maguire, M. (2001). Methods to support human-centred design. *International Journal of Human-Computer Studies*, *55*(4), 587–634. doi:http://dx.doi.org/10.1006/ijhc.2001.0503

Mao, J.-Y., Vredenburg, K., Smith, P. W., & Carey, T. (2005). The state of user-centered design practice. *Communications of the ACM*, *48*(3), 105–109. doi:10.1145/1047671.1047677

Marin, P. (2006). Potilastietojärjestelmä lääkärin apuna? *Nuori lääkäri2*, (12).

Martikainen, S., Ikävalko, P., & Korpela, M. (2010). Participatory interaction design in user requirements specification in healthcare, 304–308. doi:10.3233/978-1-60750-588-4-304

Martikainen, S., Korpela, M., Luukkonen, I., & Vainikainen, V. (2015). Where does the interaction break down ? The stakeholder map of health IT systems development and use in Finland. *Finnish Journal of eHealth and Welfare*, *7*(4).

Martikainen, S., Korpela, M., & Tiihonen, T. (2014). User participation in healthcare IT development : A developers ' viewpoint in Finland. *International Journal of Medical Informatics*, 189–200.

Martikainen, S., Viitanen, J., Korpela, M., & Lääveri, T. (2012). Physicians' experiences of participation in healthcare IT development in Finland: willing but not able. *International Journal of Medical Informatics*, *81*(2), 98–113. doi:10.1016/j.ijmedinf.2011.08.014

Mathiassen, L., Nielsen, P. A., & Pries-heje, J. (2001). Learning SPI in Practice. In *Improving Software Organizations: From Principles to Practice.* (pp. 3–22).

Mattia, A., & Weistroffer, H. R. (2010). User Participation in Successful Systems Development : A Case Study. *Proceedings of the 9th AIS SIGSAND Symposium*, 52–61.

Mayhew, D. J. (1999). *The Usability Engineering Lifecycle*. San Fancisco: Morgan Kaufmann.

McInerney, P., & Maurer, F. (2005). UCD in agile projects. *Interactions*, *12*(6), 19–23. doi:10.1145/1096554.1096556

Merriam-Webster. (2005). Merrian-Webster Medical Dictionary. Retrieved May 10, 2015, from http://www.merriam-webster.com/

Miettinen, R., Hyysalo, S., Lehenkari, J., & Hasu, M. (2003). *Tuotteesta työvälineeksi? Uudet teknologia terveydenhuollossa*. Saarijärvi: Gummerrus Kirjapaino Oy.

Miettinen, S. (2009). Designing services with innovative methods. In S. Miettinen & M. Koivisto (Eds.), *Designing services with innovative methods* (pp. 10–27). Keuruu: Otava book printings LTD.

Miettinen, S., Rytilahti, P., Vuontisjärvi, H.-R., Kuure, E., & Rontti, S. (2014). Experience Design in Digital Services. *Research in Economics and Business: Central and Eastern Europe*, *6*(1).

Miller, R. H., & Sim, I. (2004). Physicians' use of electronic medical records: Barriers and solutions. *Health Affairs*, *23*, 116–126. doi:10.1377/hlthaff.23.2.116

Minkkinen, I., & Eerola, A. (2007). Improving requirements engineering from the client" s perspective in the health care domain. In *In Proceedings of the IASTED International Conference on Software Engineering as part of the 25th IASTED International Multi-Conference on Applied Informatics (Hasselbring W, Ed.), Innsbruck, Austria*.

Money, A. G., Barnett, J., Kuljis, J., Craven, M. P., Martin, J. L., & Young, T. (2011). The role of the user within the medical device design and development process: medical device manufacturers' perspectives. *BMC Medical Informatics and Decision Making*, *11*(15). Retrieved from http://www.biomedcentral.com/content/pdf/1472-6947-11-15.pdf

MSCUI. (2015). Microsoft Health CUI - Introduction. - What is Microsoft Health CUI ? *Microsoft Health Common User Interface*. Retrieved August 8, 2015, from http://www.mscui.net/Introduction/MSCUI.aspx

Muller, M. J. (2003). Participatory design: the third space in HCI. In *Human-computer interaction: Development process* (p. 166).

Muller, M. J., & Kuhn, S. (1993). Introduction to participatory design. *Communications of the ACM*, (36).

Myers, M. D. (1997). Qualitative Research in Information Systems. *MIS Quarterly (21:2) MISQ Discovery, Archival Version, June 1997, Http://www.misq.org/supplements/. Association for Information Systems (AISWorld) Section on Qualitative Research in Information Systems, Updated Version, Last Modified: September 8, 2014*, (June), 241–242.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, *17*(1), 2–26. doi:10.1016/j.infoandorg.2006.11.001

Mykkänen, J. (2007). *Specification of Reusable Integation Solutions in Health INformation Systems*.

Mäkelä, K. (2006). *Terveydenhuollon tietotekniikka. Terveydenhuollon ja hyvinvoinnin sovellukset.* Helsinki: Talentum Media Oy.

Nebe, K., Zimmermann, D., & Paelke, V. (2008). Integrating Software Engineering and Usability Engineering. *Advances in Human Computer Interaction*, *1997*(1_2002), 331–350. doi:10.1524/itit.2002.44.1.031

Nielsen, J. (1993). *Usability Engineering*. San Diego: Academic Press, Inc.

Nielsen, J. (2001). First Rule of Usability? Don't Listen to Users. *August 5, 2001. Nielsen Norman Group Articles*. Retrieved October 26, 2014, from http://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users/

Nielsen, J. (2005). Medical Usability: How to Kill Patients Through Bad Design. *Jakob Nielsen's Alertbox*. Retrieved from http://www.nngroup.com/articles/medical-usability/

Nielsen, J. (2012). Usability 101: Introduction to Usability. *NN group articles*. Retrieved August 20, 2015, from http://www.nngroup.com/articles/usability-101-introduction-to-usability

Nielsen, J., & Norman, D. (2015). The Definition of User Experience. *NNG articles*. Retrieved August 8, 2015, from http://www.nngroup.com/about-user-experience-definition

Nies, J., & Pelayo, S. (2010). From users involvement to users' needs understanding: A case study. *International Journal of Medical Informatics*, *79*(4), e76–82. doi:10.1016/j.ijmedinf.2009.06.007

Norman, D. A. (2013). *The Design of Everyday Things* (Revised an.). The MIT Press. Cambridge, Massachusetts. London, England.

Nykänen, P., & Karimaa, E. (2006). Success and failure factors in the regional health information system design process--results from a constructive evaluation study. *Methods Inf Med. 2006*, *1*(45), 85–9.

Ora J. (2007). Lääkärin näkökulma unohtui potilastietojärjestelmistä. *Suomen lääkärilehtiSuomen Lääkärilehti*, *62*(45), 4186 – 4189.

Pekkola, S., Kaarilahti, N., & Pohjola, P. (2006). Towards Formalised End-User Participation in Information Systems Development Process: Bridging the Gap between Participatory Design and ISD Methodologies. *Information Systems*, 21–30. doi:10.1145/1147261.1147265

Pentikäinen, M. (2014). *Co-Development of Work and Information Systems An analysis of the Conxtruction of the Activity-Driven ISD methodology in 2001-2013*. University of Eastern Finland.

Petersen, L. S., Bertelsen, P., & Bjørnes, C. (2013). Cooperation and communication challenges in small-scale eHealth development projects. *International Journal of Medical Informatics*, *82*(12), e375–85. doi:10.1016/j.ijmedinf.2013.03.008

Pohl, M., Wiltner, S., Rind, A., Aigner, W., Miksch, S., Turic, T., & Drexler, F. (2011). Patient Development at a Glance : An Evaluation of a Medical Data Visualization. In *INTERACT 2011, Part IV, LNCS 6949* (pp. 292–299).

Pressman, R. S. (2009). Chapter3. Agile Development. Slide Set to accompany Software Engineering: A Practitioner's Approach. Retrieved July 4, 2015, from http://academic.brooklyn.cuny.edu/cis/sfleisher/Chapter_03_sim.pdf

Rahimi, B., Safdari, R., & Jebraeily, M. (2014). Development of Hospital Information Systems: User Participation and Factors Affecting It. *Acta Informatica Medica*, *22*, 398. doi:10.5455/aim.2014.22.398-401

Rahimi, N., & Ibarra, M. (2014). A Review of Multiple User Center Design Methods for New Product Development in Smart and Connected Health Applications, 3498–3510.

Rapoport, R. N. (1970). Three dilemmas in action research. *Human Relations*, *23*(6), 499– 513.

Reponen, J., Winblad, I., & Hämäläinen, P. (2008). Current Status of National eHealth and Telemedicine Development in Finland. *Studies in Health Technology and Informatics*, (134), 199–208.

Rogers, Y., Sharp, H., & Preece, J. (2011). *Interaction design beyond human-computer interaction* (Third.). New Delhi: John Wiley & Sons Ltd.

Rosson, M. B., & Carroll, J. M. (2002). *Usability Engineering. Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann Publishers.

Sanders, E. B. (2005). Information , Inspiration and Co-creation. *The 6th International Conference of the European Academy of Design*, 29–31.

Sanders, E. B. (2006). Design Research in 2006. *Design Research Quarterly*, *1*(September), 1–25.

Saranto, K., & Korpela, M. (1999). *Tietotekniikka ja tiedonhallinta sosiaali-ja terveydenhuollossa*. WSOY.

Scacchi, W. (2001). Process Models in Software Engineering. In *Encyclopedia of Software Engineering, 2nd Edition. John Wiley and Sons, Inc, New York, December 2001* (pp. 1–24).

Schwaber, K. (1997). SCRUM Development Process. In *Business Object Design and Implementation* (pp. 117–134). London: Springer. doi:10.1007/978-1-4471-0947-1_11

Shah, S. G. S., & Robinson, I. (2006). User involvement in healthcare technology development and assessment : Structured literature review, 500–515.

Shah, S. G. S., & Robinson, I. (2007). Benefits of and barriers to involving users in medical device technology development and evaluation Benefits of and barriers to user involvement in MDTD & E. *International Journal of Technology Assessment in Health Care*, *23*(1), 131–137.

Singh, N. P., & Soni, R. (2011). Agile software: Ensuring quality assurance and processes. *In High Performance Architecture and Grid Computing. Springer Berlin Heidelberg.*, *169*, 640–648.

Sittig, D., Kuperman, G., & Fiskio, J. (1999). Evaluating physician satisfaction regarding user interactions with an electronic medical record system. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, 400–4. doi:D005736 [pii]

Sittig, D., & Singh, H. (2011). Defining Health Information Technology–Related Errors: New Developments Since To Err Is Human. *Arch Intern Med*, *171*(14), 1281–1284. doi:doi:10.1001/archinternmed.2011.327.

Smelcer, J. B., Miller-Jacobs, H., & Kantrovich, L. (2009). Usability of Electronic Medical Records. *Journal of Usability Studies*, *4*(2), 70–84.

Sommerville, I. (2007). *Software engineering* (8th. ed.). New York: Addison-Wesley.

Standishgroup. (2013). Chaos Activity News. Monthly Newsletter for CHAOS University members. February 2013. Volume 8 issue 2. Retrieved from http://www.standishgroup.com/chaos_news/newsletter.php?id=54

Staples, M., & Niazi, M. (2008). Systematic review of organizational motivations for adopting CMM-based SPI. *Information and Software Technology*, *50*(7-8), 605–620. doi:10.1016/j.infsof.2007.07.003

Subramanyam, B. R., Weisstein, F. L., & Krishnan, M. S. (2010). User Participation in Software Development Projects. *Communications of the ACM*, *53*(3), 137–141.

Tashakkori, A., & Teddlie, C. (2003). *Handbook of mixed methods in social & behavioral research*. Sage.

Tesch, D., Sobol, M. G., Klein, G., & Jiang, J. J. (2009). User and developer common knowledge: Effect on the success of information system development projects. *International Journal of Project Management*, *27*(7), 657–664. doi:10.1016/j.ijproman.2009.01.002

Thayer, R. H., Dorfman, M., & Hunter, R. B. (2001). *Software Process Improvement*. Wiley-IEEE Computer Society Press. Retrieved from http://gov.wiley.com/WileyCDA/WileyTitle/productCd-0769509991.html

Thursky, K. A., & Mahemoff, M. (2007). User-centered design techniques for a computerised antibiotic decision support system in an intensive care unit. *International Journal of Medical Informatics*, *76*(10), 760–8. doi:10.1016/j.ijmedinf.2006.07.011

Toivanen, M., Luukkonen, I., Ensio, A., Häkkinen, H., Ikävalko, P., Jaatinen, J., … Palmén, M. (2007). *Kohti suunnitelmallisia muutoksia*.

Toivanen, M., Luukkonen, I., & Mykkänen, J. (2009). Three-level analysis for shared understanding of information systems development. *Proceedings of the IASTED International Conference*, *642*(030), 43.

Tullis, T., & Albert, B. (2008). *Measuring The User Experience. Collecting, Analyzing and Presenteing Usability metrics.* Burlinngton: Morgan Kaufmann Publishers. Elsevier,.

Urda, D., Ribelles, N., Subirats, J. L., Franco, L., Alba, E., & Jerez, J. M. (2013). Addressing critical issues in the development of an Oncology Information System. *International Journal of Medical Informatics*, *82*(5), 398–407. doi:10.1016/j.ijmedinf.2012.08.001

Vainionmäki, S., Kuusela, M., Vainionmäki, P., & Rautava, P. (2008). The quality of electronic patient records in Finnish primary health care needs to be improved. *Scand J Prim Health Care*, 117–22.

Walji, M. F., Kalenderian, E., Tran, D., Kookal, K. K., Nguyen, V., Tokede, O., … Patel, V. L. (2013). Detection and characterization of usability problems in structured data entry interfaces in dentistry. *International Journal of Medical Informatics*, *82*(2), 128–38. doi:10.1016/j.ijmedinf.2012.05.018

Wang, Y., & King, G. (2002). Philosophies and Approaches to Software Process Improvement. *SPI NEWSPAPER, European Software Process Improvement SPI and Assessments. International Software Consulting Network. Editors: ISCN LTD, ISCN GesmbH. Editing Done: 19.7.2002 ISCN*. Retrieved July 24, 2013, from http://www.iscn.at/select_newspaper/assessments/ ivf_ass.html

WebFinance. (2015). BusinessDictionary. Retrieved February 14, 2015, from http://www.businessdictionary.com/definition/user-interface.html

Wehlou. (2014). *Rethinking the Electronic Healthcare Record*. Uppsala: Man In the Middle Books AB.

WHO. (2015). Health systems. *World health organization official web site*. Retrieved June 28, 2015, from http://www.who.int/topics/health_systems/en/

Vilkka, H. (2005). *Tutki ja kehitä*. Tammi.

Winblad, I., Reponen, J., Hämäläinen, P., & Kangas, M. (2008). *Informaatio- ja kommunikaatioteknologian käyttö Suomen terveydenhuollossa vuonna 2007*. Hesinki: STAKES. Valopaino Oy.

Winter, J. (2013). *The Rocky Road: Why Usability Work is so Difficult*. Blekinge Institute of Technology.

Vredenburg, K., Isensee, S., & Righi, C. (2002). *UserCentred Design. An Integrated Approach*. Prentice Hall, Upper Saddle River.

Yen, P.-Y., & Bakken, S. (2012). Review of health information technology usability study methodologies. *Journal of the American Medical Informatics Association : JAMIA*, *19*(3), 413–22. doi:10.1136/amiajnl-2010-000020

Appendices

APPENDIX A: PRINCIPLES BEHIND THE AGILE MANIFESTO

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

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

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

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

The best architectures, requirements, and designs emerge from self-organizing teams.

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

http://agilemanifesto.org/principles.html

APPENDIX B: USER EXPERIENCE DEFINITIONS

http://www.allaboutux.org/ux-definitions

There are many definitions for user experience. Below a pool of definitions found from the literature and on the Web.

All the aspects of how people use an interactive product: the way it feels in their hands, how well they understand how it works, how they feel about it while they're using it, how well it serves their purposes, and how well it fits into the entire context in which they are using it.
- Alben (1996)

All aspects of the end-user's interaction with the company, its services, and its products. The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use. True user experience goes far beyond giving customers what they say they want, or providing checklist features. In order to achieve high-quality user experience in a company's offerings there must be a seamless merging of the services of multiple disciplines, including engineering, marketing, graphical and industrial design, and interface design.
- Nielsen-Norman Group

The overall experience, in general or specifics, a user, customer, or audience member has with a product, service, or event. In the Usability field, this experience is usually defined in terms of ease-of-use. However, the experience encompasses more than merely function and flow, but the understanding compiled through all of the senses.
- Shedroff

Every aspect of the user's interaction with a product, service, or company that make up the user's perceptions of the whole. User experience design as a discipline is concerned with all the elements that together make up that interface, including layout, visual design, text, brand, sound, and interaction. UE works to coordinate these elements to allow for the best possible interaction by users.
- UPA

User eXperience (UX) is about how a person feels about using a system. User experience highlights the experiential, affective, meaningful and valuable aspects of human-computer interaction (HCI) and product ownership, but it also covers a person's perceptions of the practical aspects such as utility, ease of use and efficiency of the system. User experience is subjective in nature, because it is about an individual's performance, feelings and thoughts about the system. User experience is dynamic, because it changes over time as the circumstances change
- Wikipedia

User Experience (abbreviated: UX) is the quality of experience a person has when interacting with a specific design.
- UXnet.org and Interaction-Design.org

A result of motivated action in a certain context. User’s previous experiences and expectations influence the present experience; this present experience leads to more experiences and modified expectations.
- Mäkelä & Fulton Suri (2001)

A consequence of a user’s internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organisational/social setting, meaningfulness of the activity, voluntariness of use, etc.)
- Hassenzahl & Tractinsky (2006)

The value derived from interaction(s) [or anticipated interaction(s)] with a product or service and the supporting cast in the context of use (e.g., time, location, and user disposition).
- Sward & MacArthur (2007)

The user experience considers the wider relationship between the product and the user in order to investigate the individual’s personal experience of using it.
- McNamara & Kirakowski (2006)
Users’ perceptions of interaction that constitute qualities of use.
- Colbert (2005)

An activity of encounter by a computer user with the auditory and visual presentation of a collection of computer programs. It is important to note that this includes only what the user perceives and not all that is presented.
- Microsoft

An umbrella term used to describe all the factors that contribute to a site user’s overall perception of a system. Is it easy to use, attractive and appropriate? Does it meet user needs?
- Public Life

The entire set of affects that is elicited by the interaction between a user and a product, including the degree to which all our senses are gratified (aesthetic experience), the meanigs we attach to the product (experience of meaning), and the feelings and emotions that are elicited (emotional experience).
- Hekkert (2006)

UX is a momentary, primarily evaluative feeling (good-bad) while interacting with a product or service.
- Hassenzahl (2008)

A person's perceptions and responses that result from the use or anticipated use of a product, system or service
- ISO 9241-210 (2010)

A set of material rendered by a user agent which may be perceived by a user and with which interaction may be possible.
- W3C

Encompasses all aspects of a digital product that users experience directly—and perceive, learn, and use—including its form, behavior, and content. Learnability, usability, usefulness, and aesthetic appeal are key factors in users' experience of a product.
- UXmatters

The design of user interaction with a system, product or service considering the usability, the enjoyment and the fit to the way users think.
- TicToc

The user experience, mostly called "customer experience" when referring to e-commerce websites; the totality of the experience of a user when visiting a website. Their impressions and feelings. Whether they're successful. Whether they enjoy themselves. Whether they feel like coming back again. The extent to which they encounter problems, confusions, and bugs.
- UsabilityFirst.com

User experience = Convenience + Design – Cost.
Convenience is the king. What makes a product convenient is quite often what makes it usable. It might also relate to the availability of the product. It might also have something to do with laziness and productivity. Defining "convenience" is by no means an easy task. As is with everything else in this chart, convenience is subjective. Design is what makes a product liked and attractive, even before it has been used. Design is what makes you want the product. It is beauty, the touch of a famous designer, a likable company, character—pretty much what brand value is thought to be.
- Nyman (2005)

The user experience is the totality of end-users' perceptions as they interact with a product or service. These perceptions include effectiveness (how good is the result?), efficiency (how fast or cheap is it?), emotional satisfaction (how good does it feel?), and the quality of the relationship with the entity that created the product or service (what expectations does it create for subsequent interactions?).
- Kuniavsky (2010)

The overall experience and satisfaction a user has when using a product or system.
- Old Wikipedia definition, still used e.g. at BitPipe.com

The overall perception and comprehensive interaction an individual has with a company, service or product. A positive user experience is an end-user‘s successful and streamlined completion of a desired task.
- Goto (2004)

UX = the sum of a series of interactions
User experience (UX) represents the perception left in someone’s mind following a series of interactions between people, devices, and events – or any combination thereof.
- Fatdux.com

User experience stands for the quality of a global experience as perceived by a person (user) interacting with a system.
- use-design.com

Users’ judgement of product quality arising from their experience of interaction, and the product qualities which engender effective use and pleasure.
- Sutcliffe (2010)

Susanna Martikainen

Towards Better Usability

Usability and End-User Participation in Healthcare Information Technology Systems Development

The usability of healthcare information systems is poor in users' opinions. To understand the reasons for this, it was examined what factors impact Healthcare IT Systems development in Finland. A number of stakeholders and factors caused challenges for HITS usability development. Recommendations and an enriched Activity-Driven Information System Development model were created based on the findings. This thesis thus contributes to the objective of progressing towards better usability of HITS.

UNIVERSITY OF
EASTERN FINLAND

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences

isbn: 978-952-61-1980-9 (nid.)
issn: 1798-5668
isbn: 978-952-61-1981-6 (pdf)
issn: 1798-5676