

Increasing User Understanding in Agile Software Development Projects with Design Practices

Henri Thor-Touch

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 27.04.2020

Thesis supervisor:

Prof. Marjo Kauppinen

Thesis advisors:

M.Sc. (Tech.) Erkkä Halme

M.Sc. (Tech.) Joanna Mehtälä



Aalto University
School of Science

Author: Henri Thor-Touch		
Title: Increasing User Understanding in Agile Software Development Projects with Design Practices		
Date: 27.04.2020	Language: English	Number of pages: 7+96
Major: Software and Service Engineering		Code: SCI3042
Supervisor: Prof. Marjo Kauppinen		
Advisors: M.Sc. (Tech.) Erkkä Halme, M.Sc. (Tech.) Joanna Mehtälä		
<p>Modern software development aims to produce valuable digital solutions by benefiting from customer- and user-centred agility. These can be supported by design practices, with which user understanding can be deepened. The goal of this thesis was to study how design practices can help small companies to increase their user understanding in agile software development projects. The empirical study was conducted as an insider action research. In this study, four design practices were tested: semi-structured interviews, user stories, scenarios, and prototyping.</p> <p>The four design practices helped to increase user understanding, by explaining who the users are, why they use the product, and how they use it. Semi-structured interviews helped to discover users' values and motivations to use the current and future versions of the product. User stories allowed for creative thinking and writing of perceived user needs in a clear sentence. Scenarios described realistic stories of users. The stories gave details of the user, their interactions, circumstances, goals, and environment. Prototyping was used alongside the other three design practices to help the users feel and test the product. Testing the product in real context allowed for spontaneous idea creation for system improvement.</p> <p>The results of this thesis indicate that a small company could use semi-structured interviews, user stories, scenarios, and prototyping to increase user understanding in agile software development projects. Increasing user understanding requires careful selection of design practices. The design practices should provide detailed information about who the users are, what their needs and motivations are, and how they would use a product.</p>		
Keywords: agile, design practice, insider action research, prototyping, small company, scenario, semi-structured interview, user-centred design, user story, user understanding		

Tekijä: Henri Thor-Touch		
Työn nimi: Käyttäjäjäymärryksen kasvattaminen suunnittelukäytännöillä ketterissä sovelluskehitysprojekteissa		
Päivämäärä: 27.04.2020	Kieli: Englanti	Sivumäärä: 7+96
Pääaine: Software and Service Engineering		Koodi: SCI3042
Työn valvoja: Professori Marjo Kauppinen		
Työn ohjaajat: Diplomi-insinööri Erkkä Halme, Diplomi-insinööri Joanna Mehtälä		
<p>Nykyaikainen ohjelmistokehitys pyrkii tuottamaan arvokkaita digitaalisia ratkaisuja hyödyntäen asiakas- ja käyttäjäkeskeistä ketteryyttä. Näitä tukevat suunnittelukäytännöt, joilla syvennetään käyttäjäjäymärrystä. Tämän työn tavoitteena oli tutkia, kuinka suunnittelukäytännöt voivat auttaa pieniä yrityksiä kasvattamaan käyttäjäjäymärrystä ketterissä ohjelmistokehityksen projekteissa. Empiirinen tutkimus toteutettiin toimintatutkimuksena, jonka toteuttaja oli diplomityön tekijä. Tutkimuksessa testattiin neljää suunnittelukäytäntöä: puolistrukturoituja haastatteluja, käyttäjätarinoita, skenaarioita ja prototypointia.</p> <p>Nämä suunnittelukäytännöt auttoivat käyttäjäjäymärryksen kasvattamisessa. Käyttäjäjäymärryksellä selitetään tuotteen käyttäjäryhmät, syyt tuotteen käytölle ja kuinka tuotetta käytetään. Puolistrukturoidut haastattelut auttoivat löytämään käyttäjien arvoja ja motivaatioita tuotteen nyky- ja tulevien versioiden käytölle. Käyttäjätarinat sallivat luovaa ajattelua ja havaittujen käyttäjätarpeiden kirjoittamista selkeinä lauseina. Skenaarioilla kuvattiin realistisia tarinoita käyttäjistä. Tarinoissa kuvattiin yksityiskohtaisesti käyttäjät, heidän vuorovaikutukset, olosuhteet, tavoitteet ja ympäristö. Prototypointia käytettiin kolmen muun suunnittelukäytäntöjen ohessa testauksen ja kokeilun tukena. Tuotteen testaus oikeassa kontekstissa mahdollisti spontaanin tuotekehitysideoinnin.</p> <p>Tämän työn tulokset viittaavat siihen, että pienet yritykset voisivat käyttää puolistrukturoituja haastatteluja, käyttäjätarinoita, skenaarioita ja prototypointia käyttäjäjäymärryksen kasvattamiseen ketterissä ohjelmistokehitysprojekteissa. Käyttäjäjäymärryksen kasvattaminen vaatii käytettävien suunnittelukäytäntöjen huolellista valintaa. Niiden tulee vastata yksityiskohtaisesti siihen, keitä käyttäjät ovat, mitkä ovat heidän tarpeensa ja motivaationsa sekä kuinka he käyttävät tuotetta.</p>		
Avainsanat: ketteryys, käyttäjäkeskeinen suunnittelu, käyttäjätarina, käyttäjäjäymärrys, pieni yritys, prototypointi, puolistrukturoitu haastattelu, suunnittelukäytäntö, skenaario, toimintatutkimus		

Preface

This journey to understand and to write about my topic started on the June 2019. Writing this thesis had taught me a lot about the topic itself, academic writing and written communication. Although this thesis is not a polished gem, despite its imperfections, I hope for some to find value in reading it.

I would like to thank our company CEO, Tuomas Kaipainen, for allowing me to do this thesis for Taito United Oy. I would also like to thank my supervisor Marjo Kauppinen for her outstanding support. Her guidance has been impeccable. Thanks to my advisors Erkka Halme and Joanna Mehtälä for their support.

To my mother and father, your son didn't become a doctor like all Asian kids, but he is now an engineer with a university degree. I am glad to have chosen this path in my studies and in my life. Much love to my family for encouraging me to attend Aalto University and for the support I have received.

To the reader, thank you for taking interest in this work. Don't fall asleep between the pages.

Espoo, 27.04.2020

Henri Thor-Touch

Contents

Abstract	ii
Abstract (in Finnish)	iii
Preface	iv
Contents	v
Abbreviations	vii
1 Introduction	1
1.1 Background and motivation	1
1.2 Research problem and questions	1
1.3 Scope of the thesis	3
1.4 Structure of the thesis	3
2 Research methods	4
2.1 Literature review	4
2.2 Empirical study	8
2.2.1 Case description	8
2.2.2 Research process	8
2.2.3 Data collection and analysis	13
3 Results of the literature review	15
3.1 Design practices	15
3.1.1 Semi-structured interviews	15
3.1.2 Task analysis	20
3.1.3 User stories	29
3.1.4 Scenarios	34
3.1.5 Prototyping	36
3.1.6 Evaluation of design choices	41
3.2 User understanding	43
3.3 Contribution of the design practices towards user understanding . . .	48
3.4 Summary of the literature review	50
4 Results of the empirical study	55
4.1 Problem diagnosis	55
4.2 Action planning	58
4.3 Action taking	61
4.3.1 Semi-structured interviews	61
4.3.2 User story crafting workshops	65
4.3.3 Scenario crafting workshops	67
4.3.4 Prototyping	68
4.3.5 Total effort of action taking	69

4.4	Evaluation of the design practices	69
4.4.1	Reflective learning	69
4.4.2	Client's evaluation and acceptance of result	72
4.5	Lessons learned from the design practices	73
5	Discussion	82
5.1	RQ1: Which design practices fit to the case company context?	82
5.2	RQ2: How do the selected design practices work?	84
5.3	RQ3: How can the design practices increase user understanding? . . .	86
5.4	Limitations of the study	87
6	Conclusions	89
	References	91

Abbreviations

Abbreviations

External User	A user representative that comes from outside of the case company or the client company
HTA	Hierarchical Task Analysis
Internal User	A user representative that belongs to the case company or the client company
IT	Information Technology
QUS	Quality User Stories
RQ	Research Question
UCD	User-Centred Design
UI	User Interface
UX	User Experience

1 Introduction

1.1 Background and motivation

Small IT businesses have often constraints in budget, resources, and schedule that may limit value creating software development decisions. This can lead to cases, where to save time and costs, design efforts for user understanding are neglected due to the high trust in market research and strong beliefs in success of the market idea. Consecutive software development iterations that are not supported by user understanding surveys, may not produce software solutions that will benefit the real end users.

According to Brhel et al. (2015), software engineering efforts in the last decade have been following either the agile or user-centred software engineering trend. In the case company, which is a small consulting company, the software development processes take some aspects from the agile and user-centered software engineering. In both software engineering approaches, understanding users better and utilizing agile tenets have been the motivation for the case company to find a way to involve user understanding research into their software development processes, in pursuit of creating valuable software solutions.

User understanding in this thesis is defined through the specification of the context of use and user requirements in user-centred design by SFS-EN ISO 9241-11 (2018) and SFS-EN ISO 9241-210 (2019). The case company had identified a need to focus more in delivering software that benefits the real end users, offers such solutions that benefit the clients, and mitigate wasted effort in software development.

Five different design practices are presented in literature review, to study their contribution towards defining user understanding in software development. Four of them are used in the empirical study: semi-structured interviewing, user stories, scenarios, and prototyping. The aim of this thesis is to provide general information of how a software engineer can use the described design practices, understand how they support in defining user understanding, and how insider action research can be used as a systematic approach to defining user understanding in software development projects.

1.2 Research problem and questions

The case company is eager to understand how different design practices, relevant in software engineering, can contribute towards understanding the context of use and user requirements. The context of use and user requirements can be seen to contain three parts: who are the users, why are they involved with the software (motivations and value), and how they act with the software (behaviour). These combined are called user understanding in this thesis. User understanding requires utilization of

design practices to define the rationale to each part of it. The research problem in this thesis is as follows:

Which design practices can help small companies to increase their user understanding in agile software development projects?

From the research problem described above, we can form three research questions (RQ) for this thesis:

RQ1: *Which design practices fit to the context of the case company agile software development process?*

In the first research question, the goal is to find a range of design practices that are suitable for the case company context. Once a range is established, a sub-group of about five design practices is selected to fit the scope of this thesis.

RQ2: *How do the selected design practices work?*

In the second research question, a description of each design practices' functionality is provided, so that this thesis can serve as a source of information for current and new, design-curious software engineers.

RQ3: *How can the selected design practices help in increasing user understanding?*

Finally, the third research question sets the theme for the empirical study on how the selected methods may help in different software development projects to increase user understanding.

To find answers to the above research questions, a combination of literature review and empirical study is used for RQ1 and RQ3. RQ2 is answered directly through literature review. Various literature sources provide good explanation and description for the design practices. For RQ1, literature gives history on design practices and their usage, and empirical study verifies whether the design practices fit to the case company context. RQ3 benefits the most from empirical study, to see the results in practice, and is supported by theoretical background from literature review.

Table 1: The ratio between the use of literature review and empirical study in answering the research questions.

	Literature Review	Empirical Study
Research Question 1	Low	High
Research Question 2	High	-
Research Question 3	Low	High

1.3 Scope of the thesis

This thesis focuses in understanding design practices that fit to the context of the case company and contribute to the increase of user understanding in agile software development projects. This was studied through a case project for a client. A group of five design practices were chosen for this thesis: semi-structured interviewing, task analysis, user stories, scenarios, and prototyping. Only task analysis was not used in the empirical study.

Also, this thesis provides the author's definition for user understanding, and how the design practices contribute towards increasing or validating user understanding. In creating the definition for user understanding, the basis for the definition comes from the SFS-EN ISO 9241-11 (2018) and SFS-EN ISO 9241-210 (2019) standards, as well as several literature sources describing cognitive aspects, such as emotions, motivation, perceived value and value creation, and user needs.

Results from the empirical study cannot be generalized, but the literature review of the design practices provide general information about them.

1.4 Structure of the thesis

This thesis contains 6 chapters. In the first chapter, the author provides the introduction to the thesis, explaining the motivation, research problem and questions, and the scope. Chapter 2 touches the research methodology used in this thesis, how the literature review and the empirical study were conducted. Chapter 3 provides a description to each design practice and answers to the RQ2. In addition, the author gives his definition of user understanding, the contribution of each design practice towards eliciting or creating user understanding is described, and a process for conducting user understanding studies according to the insider action research tenets is proposed in the final subsection.

Chapter 4 presents the results from the empirical study. It explains the five parts of action research: problem diagnosis, planning of the action to be taken, executing the planned actions, evaluating the outcomes, and specifying the lessons learned from the study.

Then, in chapter 5, the results to each research question are discussed and the limitations of this study are mentioned. In the final chapter 6, the author gives his conclusions to the thesis work.

2 Research methods

2.1 Literature review

The literature was searched by using Google Scholar and Aalto University Library's search engine. The selection of literature was based on two criteria: First, the design practices would be chosen together with the thesis advisors, who are part of the same organization as the author. Second, general literature were chosen based on their matching with the keywords used in this thesis. Table 2 presents the initial candidates for this thesis. Tables 3, 4, 5, 6, 7, and 8 present the keywords in question for each design practice and user understanding.

Table 2: A group of possible design practices for this thesis. The author and the advisors picked 5 design practices to be in the scope of this thesis.

Design practice	In this thesis
Affinity diagram	
Analysis of existing documentation	
Apprenticing	
Brainstorming	
Card sorting	
Domain analysis	
Ethnographic methods	
Focus groups	
Heuristic evaluation	
Interaction design	
Iterative design	
Introspection	
Joint Application Design workshops	
Laddering	
Protocol analysis	
Prototyping	X
Questionnaire	
Rapid Application Development workshops	
Repertory grids	
Scenarios	X
Semi-structured interviewing	X
Structured interviewing	
Surveys	
Task analysis	X
Unstructured interviewing	
Use cases	
User stories	X
Work modelling	

Defining the scope of this thesis began with choosing 5 design practices. The author had prepared a short introduction about each listed design practice to the advisors. Then, the advisors gave their ranking to each of the listed design practices. The ranking was based on the advisors' evaluation of how each design practice would fit to the cace company's current software development processes. The highlighted design practices in the previous Table are the top 5 design practices which were chosen for this thesis.

The literature sources for the selected design practices was searched by using certain keywords or by the author's knowledge about relating papers. Tables 3, 4, 5, 6, 7, and 8 present the literature used in literature review and the keywords or rationale for searching the sources. GS/G/LS/P present how the literature sources were searched: Google Scholar Search (GS), regular Google Search (G), Aalto University Online Library Search (LS), and visiting Aalto University library and searching through the books physically (P).

Table 3: Selected publications related to semi-structured interviews.

Literature source	Keywords/Rationale	Type
Adams (2010)	<i>semi-structured, interview, challenges</i>	GS
Beyer and Holtzblatt (1998)	<i>holtzblatt</i> . The author was given a tip about Holtzblatt from an UCD professor	LS
Bryman (2016)	Found by visiting Aalto University library and looking up social research or qualitative research methods	P
Hackos and Redish (1998)	<i>task, analysis</i> . Mentioned in Kujala and Kauppinen's (2004) paper	LS
Kujala and Kauppinen (2004)	<i>selecting, users, for, user-centred, design</i>	GS
Kvale (1996)	<i>interviews</i>	LS
Kvale (2007)	<i>doing, interviews</i>	LS
Rabionet (2011)	<i>semi-structured, interview, in, design</i>	GS
Wilson (2013)	<i>semi-structured, interview</i>	LS

Table 4: Selected publications related to task analysis.

Literature source	Keywords/Rationale	Type
Annett (2003)	<i>hierarchical, task, analysis</i> . This paper was found using regular Google search, belonging to an Aalto University's course as course material.	G
Courage et al. (2008)	Found in Aalto University library when browsing books about design in software engineering	P
Crystal and Ellington (2004)	<i>task, analysis, techniques</i>	GS
Hackos and Redish (1998)	<i>task, analysis</i> . Mentioned in Kujala and Kauppinen's (2004) paper	LS
Hornsby (2010)	<i>hierarchical, task, analysis</i>	G
Holzinger (2005)	<i>task, analysis, usability</i>	GS
Norman (1988)	The author knew about the book	P
Norman (2013)	The author knew about the book	P
Ritter et al. (2014)	<i>user, centered, design, users</i>	LS
Vredenburg et al. (2002)	<i>user, centered, design, in, industry</i>	GS
Stanton (2006)	<i>hierarchical, task, analysis</i>	GS
Zowghi and Coulin (2005)	<i>requirements, elicitation</i>	GS

Table 5: Selected publications related to user stories.

Literature source	Keywords/Rationale	Type
Cohn (2004)	<i>user, stories, applied</i>	LS
Lucassen et al. (2016)	<i>user, story, quality</i>	GS
Deemer et al. (2012)	The author knew about the publication	G

Table 6: Selected publications related to scenarios.

Literature source	Keywords/Rationale	Type
Alexander and Maiden (2004)	<i>scenarios, in, system, development</i>	GS
Bardram (2000)	<i>scenario, based, design</i>	GS
Bødker (1999)	<i>scenarios, in, design, work</i>	GS
Carroll (1999)	<i>scenario, based, design</i>	GS
Carroll (2002)	<i>scenarios, in, design, work</i>	GS
Carroll and Rosson (1990)	<i>scenario, design, human, computer, interaction</i>	GS
Cross (2001)	Referenced in Carroll (2002)	GS
Interaction-design.org (2019)	<i>scenario, design</i>	G
Rosson and Carroll (2008)	Found in Aalto University library when browsing books about design in software engineering	P

Table 7: Selected publications related to prototyping.

Literature source	Keywords/Rationale	Type
Beaudouin-Lafon and Mackay (2008)	Found in Aalto University library when browsing books about design in software engineering	P
Buchenau and Suri (2000)	<i>prototyping</i>	GS
Camburn et al. (2017)	<i>prototyping, methods, and, techniques</i>	GS
SFS-EN ISO 9241-210 (2019)	The author knew about the publication	LS
McElroy (2016)	<i>prototyping</i>	LS
Warfell (2009)	<i>prototyping</i>	LS

Table 8: Selected publications related to user understanding.

Literature source	Keywords/Rationale	Type
Almquist et al. (2016)	The author knew about the publication	GS
Cerejo (2018)	<i>how, to, understand, users</i>	G
Gautam (2012)	<i>user, motivations, and, emotions</i>	G
SFS-EN ISO 9241-11 (2018)	The author knew about the publication	LS
SFS-EN ISO 9241-210 (2019)	The author knew about the publication	LS
Kauppinen et al. (2009)	<i>value, creation, kauppinen.</i> The author knew about the publication	GS
Maslow (1943)	<i>maslow, 1943</i>	GS
McLeod (2018)	<i>maslow, hierarchy, of, needs</i>	G

2.2 Empirical study

2.2.1 Case description

Taito United Oy is a Finnish IT consultancy company that was founded in 2012. It has about 30 employees at the time of this thesis. At the case company, various software development projects are done for various clients. The clients often have a need for a whole digital solution for their business, which the case company designs, develops, and delivers. The company is located in Espoo at Spektri Business Park.

At the case company, software development follows agile software development tenets, many of which are from the agile manifesto:

- individuals and interactions over processes and tool,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation,
- and responding to change over following a plan (Beck et al., 2001).

Following these tenets, the case company uses modified Scrum and Kanban practices to achieve software development agility and design agility.

The case company is searching for a set of design practices, which increase user understanding, to add into their agile software development process. A case project was selected for testing the design practices. The project was a digital coaching platform, done as a co-project with a client company. The client company was a small company located in Espoo as well. At the time of the empirical study, the digital coaching platform was in further development.

2.2.2 Research process

The research process follows the insider action research process. In insider action research, qualitative research is conducted by combining both practice and research to find ways to e.g. provide systems improvement, organizational and self learning, and refinement of current theory (Avison et al., 1999; Coghlan and Brannick, 2005; McNiff, 2013). The base iterative process of insider action research contains three distinct parts: problem diagnosis, action intervention, and reflective learning (Avison et al., 1999). The insider action research used in this thesis contains five parts. Coghlan and Brannick (2005) divide action intervention into two distinct parts: action planning and action taking. Reflective learning is also divided into two parts, evaluating of action taken (Coghlan and Brannick, 2005), and specifying learning, which focuses on reflection and provision of self- and organizational learning. Figure 1 presents the action research process used in this thesis.

The insider action research was guided by the need of mapping the current and wanted user understanding. User understanding explains the user groups, needs

and motivations for product use, and how the product is being used. The following questions help in providing detail to user understanding: *who uses the product, why they use it, and how they use it?*

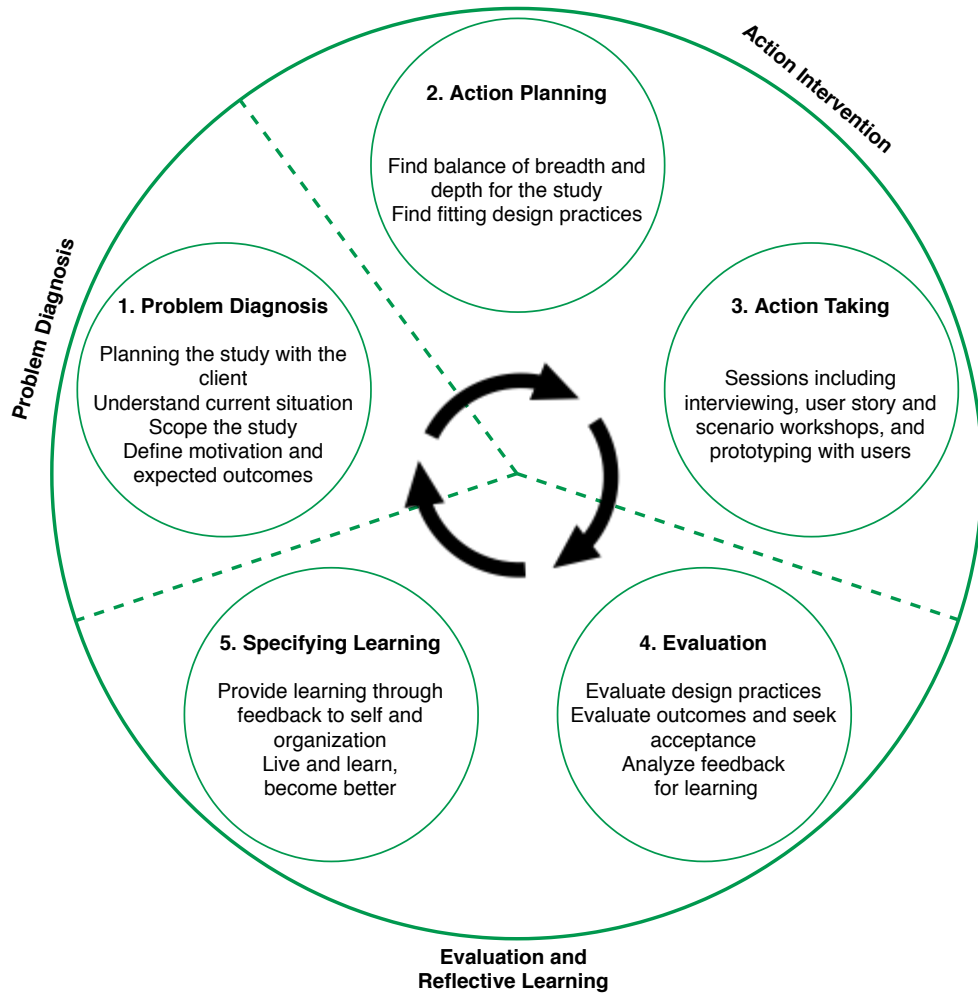


Figure 1: The insider action research process used in this thesis. Action intervention is disseminated into action planning and action taking, as proposed by Coghlan and Brannick (2005). Evaluation and reflective learning has two parts, evaluating the design practices and the outcomes, and specifying learning for improvement.

The insider action research began with problem diagnosis. **In problem diagnosis**, the author specified the motivation for the study and the motivation for mapping the current and wanted user understanding with the client. Specifying the motivation meant understanding the needs for insider action research, which can be e.g. a system improvement, organizational learning, change management, or self-learning (Coghlan, 2001; McNiff, 2013). From the specified needs, forming the goal, purpose of the study, or specifying the desired outcomes can mitigate the risk of producing meaningless results and provide means for evaluating the success or failure of the study. Lastly, the author agreed with the client the evaluation criteria for acceptance of results. Explicit evaluation criteria is important to agree prior to moving into

action planning (Avison et al., 1999), as those can aid in the selection of methods and practices for action intervention and reflective learning.

It should be noted, that the author had documented the results from each part of the insider action research. Avison et al. (1999) say that the insider action researcher must document the research process thoroughly to have documentation at hand for self-reflection and learning. Action planning followed next.

In action planning, the author planned on choosing the relevant actions for execution together with the client. The decisions were affected by the input from the client about the breadth and depth they wanted for the study, and the time and budget. As those were decided, the author gave his suggestion for forming the plan to the study. When planning the action taking, the researcher must explicitly describe the reasoning for the selected actions (Avison et al., 1999; McNiff, 2013). After the plan was formed, the author proceeded to recruiting participants for the study according to the planned breadth, depth, schedule, and budget. Action taking followed next.

Action taking followed action planning. It focused on executing the planned design practices (semi-structured interviews, user stories, scenarios, and prototyping) with the selected schedule and participants. The plan included having a semi-structured interview session for all and a workshop with differing content depending on the participant. The results from conducting the actions were collected for analysis, evaluating the success of the study, and specifying learning from the conducted design practices.

Table 9: The design practices and their participants.

Practice	Participants
Semi-structured interview	User 1 User 2 User 3 Project Manager Product Owner
User story crafting workshop	User 1 User 2 User 3
Scenario crafting workshop	Project Manager Product Owner
Prototyping	User 1 User 2 User 3 Project Manager Product Owner

After action taking, the study moves on to evaluating the results. **In evaluation**, the author and the participants evaluated the design practices and the outcomes of each session. Evaluating the design practices yielded useful information on how the selected practices had contributed towards system improvement and change. Outcomes of the sessions provided evidence for evaluating the success or failure of a session: low outcome of e.g. user stories could signal of a wrong type of user or audience participating in the user story crafting sessions. The client participated in the evaluation of the outcomes to give acceptance of the study outcomes. Acceptance was evaluated against the defined motivation for user understanding and the expected outcomes of the study.

Finally, the last part of the insider action research process used in this thesis, is specifying learning. **Specifying learning** focused in summarizing the results from action taking and design practice evaluation, in order to specify improvements for the conducted study and future studies. Avison et al. (1999) state that action research focuses in change and reflection, which can be understood as finding improvements. Coghlan and Brannick (2005) also highlight the importance of improvement in evaluating action. The knowledge generated through reflection and further refined into learned lessons helped in the study to understand better how the academic based insider action research process fit to the current case company processes, and how to improve.

It should be noted that some challenges exist with action research. Action researchers are often tied closely to the project they are researching, thus they must deal with preunderstanding, which may affect attitudes and choices towards and during the research process. Preunderstanding is a term used to describe existing knowledge of the research target the researcher has before starting the research, or the process of building on the closeness the researcher has with the domain and at the same time creating distance from it for objective inspection (Coghlan, 2001; Coghlan and Holian, 2007). A challenge of balancing objectivity and subjectivity. Coghlan (2001) notes about preunderstanding, that manager-researchers (which the author represents as a project manager in his organization) "may assume too much and so not probe as deeply as if they were outsiders or ignorant of the situation". Furthermore, "they may think they know the answer and not expose their current thinking".

Another challenge is role duality, which may create role confusion, conflict, or overload to the researcher (Coghlan and Holian, 2007). Role duality may also lead to political problems within organizations, and contribute towards research bias (Coghlan, 2001), if organization membership affects or endangers the objectivity of the researcher.

Coghlan and Holian (2007) suggest that insider action research projects can provide good grounds for an excellent academic thesis, but at the same time they warn that for students, the management of time, revenue, and credibility can be at stake when conducting insider action research to one's organization. They continue, that a 'failed' insider action research project may result in limitations to career options.

One possibility for mitigating the damage from failure is to manage expectations and be open about risks involved with the insider action research. Creating an open environment and building trust

These challenges can be difficult to overcome and definitely require experience to identify these challenges during research work and to react accordingly. Proper planning and designing of the study process can help in minimizing the occurrences of above mentioned challenges, if fitting literature or coaching is available. However, the lack of experience can be a limiting factor, that affects the effectiveness of foreseeing possible problems.

But, despite the aforementioned challenges, insider action research can be successful with proper execution. Patton (2002) suggests for researchers tackling with objectivity and subjectivity to attain *a stance of neutrality* in qualitative research work. He defines *neutrality* as an attitude towards research, that can be described as such: the researcher "does not set out to prove a particular perspective or manipulate the data to arrive at predisposed truths" and entering the research area without a strong personal opinion, a need to prove any particular theory, and supporting any predetermined results. *Empathic neutrality* focuses on neutrality with human contact; support to openness and being nonjudgmental when interacting with other person's thoughts, emotions, and behaviour (Patton, 2002). E.g. in an interview situation, *an empathically neutral* person carrying an inquiry can be perceived as someone who is caring about and interested in the people they are inquiring, but neutral to what they disclose (Patton, 2002).

For the author, balancing between enough objectivity and subjectivity during insider action research was a challenge due to the lack of experience in conducting insider action research. In attempt to counter researcher bias, the author did his best to conform to Patton's (2002) concept of neutrality. In practice, the author attempted to maintain an open attitude towards the research process, not judging any of the steps required in insider action research, and building trust with all relevant stakeholders by being honest about the process, the risks involved, and the expected outcomes. Continuing on honesty, the data collected was not altered. This course of action may have increased the risks of having credibility at stake or career limitations, if the results would not be pleasing.

In case of a failure of the study, all findings and reasons must be documented and reported properly to provide self and organizational learning of the event. To mitigate any backlashes that may had followed a failed insider action research, the researcher should be ready to provide evidence of neutral approach to research and provide reasons in sufficient depth. If there was a risk of failure due to incompetence or lack of experience, the risk should be communicated as soon as possible with the relevant stakeholders to maintain trust and to find best practices to mitigate the risk. Patton (2002) says about empathic neutrality that the researcher should be perceived as someone who is caring about and interested in the project and doing her best to help it, so that failure of the research can be perceived in a more positive light than what it could have been.

The author, at the time of writing this thesis, was a 26 years old Master's student at Aalto University with academic knowledge about software development and engineering, software testing and quality assurance, requirements engineering, usability evaluation, user experience research, and service design. His study track is Service Design and Engineering in the major of Software and Service Engineering in Computer, Communication and Information Sciences. During his education, he had used in practice semi-structured interviewing, user stories, scenarios, and prototyping on various Master level courses. In usability evaluation, understanding the user flows within a system through heuristic evaluations and rapid ethnography studies have been the closest the author has done to task analysis work. The author has been in the case company as a project manager and a software consultant since 01.06.2018.

2.2.3 Data collection and analysis

The author had prepared a presentation for the client to explain the concept of the study, length, benefits, and opportunities. The introductory presentation was given face-to-face. Agreement for the study was informal, as the author and the client agreed to conduct the study verbally. This was possible due to the high level of trust between the author and the client. Decisions were documented into personal notes and later refined into notes for the client, sent via e-mail.

In problem diagnosis, the results from charting the current state were written down into personal notes during a meeting. In the same way, decisions and plans for next actions were documented into personal notes. All personal notes would then be refined into relevant wholes either into a separate text document or included into PowerPoint presentations used in meetings with the client. The documents would then be sent via e-mail. Sending summaries of meetings through e-mail provides couple benefits: the author was forced to filter unnecessary information and the e-mails worked as written proof of past activities.

For action planning, agreement for author's proposal was noted down into personal notes. The report of action planning results included notes of the estimated length of the coming action taking, how many hours of effort would be expected from each participant, how many participants would there be, and from where would the participants be recruited from. These notes were in a form of a PowerPoint presentation. Changes to proposal would be made during the meeting and the finalized and accepted proposal made together with the client would then be sent to them.

In action taking, the author held two sessions per participant: first, an interview sessions and then a workshop session. In the interview sessions, all of them had voice recorded by the author's phone. The author used a OnePlus 6 phone with no modifications and using the voice recorder app of the phone. During the interview, the voice recording was left running after the formal part of the interview was finished, so that during informal feedback discussion, additional information could be captured

on tape. In addition to recording, a personal notebook was used to note down important highlights. After the interviews, recordings were transcribed. Answers from interviews were categorized under their respective questions and grouped.

The workshops had practical tasks which involved writing either user stories or scenarios. For user stories, the author had created to his personal Google Drive a Google Sheet document, which the author shared to the participant and into which the participant wrote their user stories on their laptop. For scenarios, data was collected similarly, with the exception of a Google Sheet document being a Google Docs document, which was more suitable for writing long texts. Prototyping was used in both cases by showing the current functioning implementation of the product and going through relevant features of the admin panel from the author's laptop. From scenarios, the author interpreted clear user stories and added them to the other user stories. All user stories would then be grouped under relevant predetermined categories, which were provided by the client and were used in the workshops. Similar user stories were not filtered so that the client could see if something was proposed multiple times, suggesting an importance for such feature. The results were then summarized into a PowerPoint presentation, that was shown and sent to the client.

In evaluating the results, the focus was to inspect the results from action taking with the client and evaluate whether they contributed towards the motivation of the study, which was finding user requirements that focus on improving the current admin panel of the product. Then, the used design practices and methods were evaluated based on feedback collected from participants at the end of each session to judge the agility and conformance to the case company context. The author's evaluation consisted of personal judgment and participant feedback, whether such practices could be conducted light-weighted and iteratively alongside software development, as they were done in action taking. The case company has many software development cases for its clients, which means that fitting the company context requires for the design practices to assist the company's software development efforts without hindering it.

For specifying learning, the author reflected on the results of the study, feedback from participants, and his experiences to find results into the following questions:

- What went well or did not go well?
- How could the process be improved for future cases?
- How can the case company and the client company learn from this study?
- What did I as a researcher and as a participant in this project learn from this experience?
- Could this process work alongside our agile software development projects?

The results from self-reflection was documented into a PowerPoint presentation for sharing the self- and organizational learning to relevant stakeholders.

3 Results of the literature review

3.1 Design practices

3.1.1 Semi-structured interviews

Interviews are often used in qualitative research to capture various types of data that is not always quantifiable (Adams, 2010; Rabionet, 2011; Wilson, 2013). The semi-structured interview is but one method amongst many other interview types that are used in qualitative research. It combines elements from both structured and unstructured interview method, hence the name semi-structured, taking advantage of predefined questions used in structured interviews that form the red line of the interview, and the use of probe questions used in unstructured interviews that promote exploration of a topic at each question (Wilson, 2013). As a qualitative data analysis tool, semi-structured interviewing is a tool towards gathering data on users and deepening user understanding (Hackos and Redish, 1998).

Wilson (2013) gives good knowledge about interview techniques associated with semi-structured interviews in his book. Semi-structured interviews allow interviewers to discuss issues with users and delve deep with the discussion topics thanks to probe questions used in the interview situation, according to him. Further, users can guide the conversation and present their opinions and ideas in more depth depending on how much the interviewer allows the user to steer the discussion (Adams, 2010). However, there should not be a fear of derailing discussion, as a well designed semi-structured interview should contain mechanisms to steer the conversation back on topic when necessary through probes and prompts in the structure of the interview (Wilson, 2013).

As for the structure of semi-structured interviews, we can view it from two viewpoints: First, the structure of the interview procedure, and second, the structure of the interview itself. For the interview procedure, we can divide it to three distinct parts: (1) Preparation for the interview, (2) Conducting the interview, and (3) After the interview. Each part is explained through suggestions and notes from the following authors: Hackos and Redish (1998), Adams (2010), Rabionet (2011), and Wilson (2013). The interview structure is presented in the preparation part.

Preparation for the interview. Before an interview can happen, it must be planned and thought out before it can be executed. Wilson (2013) suggests forming the goal or research purpose of the interview which will help in making decisions for the context and contents of the interview. Further, he provides a few examples for general goals for semi-structured interviews:

- Exploring a particular topic, problem, or issue
- Understanding how a process or function works
- Understanding how particular groups in an organization work together

- Determining what is efficient and inefficient about particular work flows
- Gathering background material for creating personas, task models, or other artifacts
- Testing ideas or hypotheses from other sources
- Confirming (or disconfirming) results from other methods.

When the goal for the interview is set, one should consider what questions to ask and which users to interview to reach the desired goal or research purpose. For interview questions, one could consider including general and specific questions (Hackos and Redish, 1998). General questions are good for opening up a topic and having specific questions helps directing the conversation. Also, the interviewer should have a handful of *ad hoc* questions available that can be triggered at any time during the discussion to delve deeper on a topic (Wilson, 2013). When thinking about questions, consider the following (Wilson, 2013):

- Avoid the tendency to add “interesting” questions that do not relate to a research goal. Ensure that each question is relevant to the goals or hypotheses of your project. It should be possible to connect each question to a clear business or research goal. If you can’t connect a question to a clear goal, then delete it
- Avoid questions that are overly long or complex. You can use probes to get more details and clarify responses
- Don’t ask double questions such as “How would you describe the usability and reliability of the new software?” Break questions like this into two questions
- Use language that is appropriate for your participants. Adapting your language to match the participant enhances your credibility and minimizes the risk of you looking foolish in an interview situation. If the participant uses often terms that you are not familiar with, look them up in preparation for other interviews
- Review the question order for obvious biases and sensitivities. You may not want to ask a threatening initial question that might influence later responses.

There is a good list of example general questions in Wilson’s book. If the the interview focuses on a set of users that need to be known before thinking about the interview questions, Hackos and Redish (1998) suggest brainstorming a preliminary list of users and creating a user/task matrix to create a basis for the interview study.

As for how many users you should interview, most optimal would be to "interview as many subjects as necessary to find out what you need to know" (Kvale, 1996). But, one should be vary of resources and time available for conducting interviews (Kujala and Kauppinen, 2004). In addition, they state that evaluating the number of participants is difficult. But, one could start small and aim for breadth of representation in users, starting with 5 or 6 to 10 users (Hackos and Redish, 1998; Beyer and Holtzblatt,

1998). About the participants, Wilson (2013) suggest to avoid over-recruiting from one company if that can be avoided, so that one does not miss problems from other customers or users.

The selected questions and prompts form the basics of a semi-structured interview. But, the questions and prompts alone cannot form a good interview basis without being structured properly. Hackos and Redish (1998), Kvale (2007), and Wilson (2013) suggest creating an interview guide with the basic questions and having a script for the interview. Moreover, Wilson (2013) suggests to order the question from easier to more complex questions.

Table 10: A basic semi-structured interview guide by Wilson (2013).

Activity	Comments/Questions	Approximate Time
Introduction	Brief the participant. Introduce self. Explain goals of interview. Review interview method, use of data, confidentiality, so on.	10 min
Structured topics	Topic 1: Background Question 1a Probe 1 Probe 2 Probe 3 Topic 2: Context of Work Question 2a Probe 1 Probe 2 Question 2b Probe 1 Topic 3: Use of Product Question 31 Probe 1 Topic N: Additional topics	40 min
General questions and open dialogue with participant		30 min
Closing comments and completion of any paperwork (receipts, final questionnaire, etc.)		10 min

When the structure of an interview is designed, the interviewer should also check whether the following are set for the interview: location, recording tools, ethics, and any necessary documents related to the interview. The location of the interview affects the privacy of the interview, the pleasure of environment, the relaxation of participant (Adams, 2010), and the convenience of travel to the interview (e.g. considering an interview location that is easily accessible and is within reasonable distance for the participants increases convenience).

Recording tools are recommended for an interview. Sometimes, the cognitive capabilities of the interviewer and the length of the interview do not allow for 100% accuracy of memory for the interview contents. Various tools and methods exist for taking notes. One could take notes during and/or after the interview (Rabionet, 2011; Wilson, 2013), with tools such as pen and paper (Adams, 2010), an audio recorder, a video recorder (Adams, 2010; Rabionet, 2011; Wilson, 2013). Sometimes recording at an interview is not allowed, then, having a partner with you can help in memorizing details of the interview (Wilson, 2013). Furthermore, a partner can help with note taking and operation of the recording tools, but, beware of the ratio between participants and interviewers as a single participant might feel intimidated with two interviewers (Wilson, 2013). On electronic tools, Adams (2010) notes to make sure they are charged and have means of having more energy when necessary (e.g. replacement batteries).

One should take into account ethical issues related to interviews. Kvale (1996, 2007) outlines several aspects of interview ethics, of which in this thesis we will be looking into *informed consent*, *confidentiality*, and *consequences*. Informed consent is a piece of information that outlines the overall purpose of the interview, the main features of the interview design, the possible risks and benefits for participation, and the participants consent on participation with the right to withdraw from it at any time (Kvale, 1996, 2007). Further, with minors or representatives of a company, it is important to consider who should give the consent.

Confidentiality of participants should be protected, if they wish so or the interview results in publication of identifiable data of the individual. Important considerations are to think of who will have access to the interview material, a written agreement about disclosure of interview data if identifiable data would be published to third parties, and anonymity of participants in the event of disclosed information that can raise legal issues or contain malicious information to protect the interviewer (Kvale, 1996, 2007).

Consequences of an interview study include both positive and negative consequences. While negative consequences exist, one should always strive for positive consequences if possible, minimizing the risk of harms to the participant. In other words, following the principle of beneficence (Kvale, 1996). The interviewer should think of the consequences that may occur for the participant, for the larger group the participant belongs to, or the consequences from disclosed information from the participant that they may later regret (Kvale, 2007).

Finally, as the last steps for preparing for an interview, Wilson (2013) suggests piloting the whole process of the interview from start to finish and preparing all necessary documents. Piloting helps to verify the time estimation and the flow of the interview process, and creates opportunities for interview process refinement. For the interview documents, the interviewer should prepare according to the need, the interview guide, an informed consent form, a Non-Disclosure Agreement (NDA), a demographic survey, and/or receipt for compensation for participating (Wilson, 2013).

Conducting the interview. The social and conversation skills of the interviewer affect a number of factors that contribute towards the outcome of an interview. For social skills, the interviewer should take approach the participant in a calm, polite and professional manner with emotional control (Adams, 2010; Wilson, 2013). Consider the positioning of yourself, the interview space, and possible equipment positioning before the start of the interview discourse and make adjustments as necessary with the participant (Wilson, 2013).

Then, Wilson (2013) suggests starting the interview with an introduction to the interview process. About the process, he proposes for the interviewer to consider the following:

- Give a brief introduction about the interview topics and goal, the stages of the interview process, recording and ethical issues, and the commenting and prompting during conversation
- The amount of time the interview takes
- What the interviewer will do when the participants needs to answer a phone or leave momentarily
- What will the interviewer do with the data and whether the participant gets access or a copy of the transcribed/analyzed data.

The conversation skills of the interviewer affects how well the participant elaborates on interview topics. Adams (2010) points out the following conversation skills that help the interview situation: listening carefully, managing silences, being non-judgmental, and allowing the participant to guide. Listening to the participant allows interview topics to be explored in sufficient detail, not moving to another topic too quickly, and creating possibilities for using prompt questions for more in-depth exploration, with keeping in mind the time constraints and staying in topic (Adams, 2010). Managing silences is an art of giving space to the participant and providing room for more conversation (Adams, 2010). Silence during conversation allows for the participant to reflect on their answers and possibly break the silence by adding more to the conversation topic (Kvale, 2007). But, leaving the participant in prolonged silence can signal negative attitudes, such as disapproval or absence of understanding, to which the interviewer has to make the call to go forward when no further answers can be obtained on a particular topic (Adams, 2010; Wilson, 2013). Being non-judgmental towards the participant's views and remarks on certain topics is important to allow

the participant to express details natural to their language and attitude towards the interview topic (Adams, 2010). Moreover, if the interviewer shows judgmental attitude towards the participant or challenges their statements immediately, they might feel offended and become less responsive, or even end the interview prematurely (Adams, 2010). Allowing the participant to guide the conversation allows exploration of the interview topics in their terms and can potentially provide more details than the originally planned script. When allowing such control, it is important for the interviewer to be non-directive by giving affirmative responses verbally or through body language (Adams, 2010). The body language during an interview can provide hints of the interviewer's or participant's attitude towards the other person, the topic, or the interview itself. For example, the interviewer showing tiredness or yawning can signal lack of interest towards what the participant has to say in the interview Wilson (2013).

When the interview is ending, signal a clear end to the interview by thanking the participant, putting away note-taking equipment and proceeding to handing out any promised compensation (Adams, 2010). He notes that the interviewer should be aware when the official interviewing stage is over, as the participant might engage the interviewer in a post-interview conversation. Thus, Bryman (2016) suggests to not stop recording as the interview is winding down and to keep recording until the last possible moment to capture any rich post-interview conversation. Also, the interviewer can at the end of the interview ask the participant to answer a demographic survey if necessary (Wilson, 2013).

After the interview. Adams (2010) and Wilson (2013) suggest to write notes after the interview. The notes can discuss, for example, the location of the interview, any interesting events during the interview, most memorable claims or views, or anything that could have affected the interview (lack of privacy, manager moving nearby, etc). Anything the interviewer deems noteworthy should be noted down (Adams, 2010). Wilson (2013) suggests to book enough time for analysis of the interview notes before the next interview, to break down any interesting aspects or topics that could be brought up with the next interview participant. Furthermore, it is important to consider tools for transcribing the interview, so that analysis can be based on other than the interviewer's memory, in case recording devices were used in the interview. Transcribing interviews can result in masses of data that takes time to analyze. Bryman (2016) suggests to consider transcribing only the interesting parts of an interview, as an interview most likely contains data that is irrelevant to the study goal.

3.1.2 Task analysis

Designing successful interfaces and writing successful documentation requires knowledge about your users and their work (Hackos and Redish, 1998). They emphasize in discovering user's actions and goals, capturing the real objectives and action that users express better during a demonstration of their work rather than telling about

it in an interview. Further, as time goes by with a certain work or a product, users change over time and different level of users behave differently when doing the same tasks or working with a same product.

Task analysis aims to understand user's tasks and goals by using various inspection methods, such as interviewing, observing, shadowing users, and doing day in a life of studies (Hackos and Redish, 1998; Courage et al., 2008). Courage et al. (2008) suggest three important analyses: (1) the user's tasks, (2) the user itself and (3) the user's environments (Courage et al., 2008).

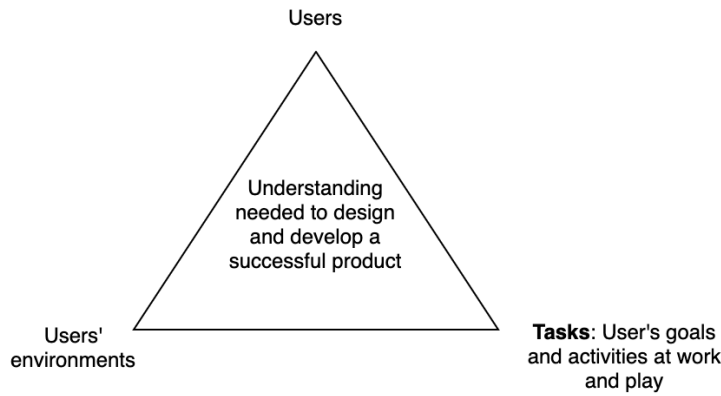


Figure 2: Designing and developing a successful product requires knowledge about users, their environment, and tasks Courage et al. (2008).

Table 11: Questions for the user and environment analyses (Courage et al., 2008).

Analysis	Questions
User analysis	<p>Who are the users?</p> <p>What user characteristics are relevant to what you are designing?</p> <p>what do the users know about the technology?</p> <p>What do the users know about the domain?</p> <p>How motivated are the they?</p> <p>What mental models do they have of the activities your product covers?</p>
Environment analysis	<p>What is the physical situation in which the work occurs?</p> <p>What are the technologies available to the user?</p> <p>Social, cultural, and linguistic considerations:</p> <p>What will make the new acceptable in the users' world?</p> <p>How will the new product change the users' world?</p> <p>How will you help the users make the transition from the old world to the new world?</p>

Understanding the user's tasks brings us closer to understanding goals and behaviour associated with a user's work or leisure, which is the focus of task analysis. Understanding users themselves means user analysis, "learning about ordinary users by observing them in action" (Hackos and Redish, 1998; Courage et al., 2008). Analysis of the user environment involves understanding their physical, technological, cultural, social, and political environments that influence their work (Hackos and Redish, 1998; Courage et al., 2008).

Courage et al. (2008) set four principles for task analysis:

1. Task analysis is an integral part of a broader analysis that includes understanding users and their environments
2. Task analysis includes understanding users' goals
3. Although the focus, methods, granularity, and presentation of information may differ at different times, task analysis is relevant at all stages of the design and development process
4. The practical reality is that task analysis for given project depends on many factors.

Task analysis is an integral part of a broader analysis. Courage et al. (2008) state that task analysis alone will not give you enough information to design or evaluate a product. From that perspective, they propose that task analysis is just one corner of a triangle in the effort of triangulating data to produce successful design. The other two analyses are user analysis and environment analysis (see Figure 2). When conducting these analyses, they suggest asking the questions, in Table 11, about users and their environment.

Task analysis includes understanding user's goals. Users complete tasks to achieve a goal and different users may take different tasks despite having the same goal (Hackos and Redish, 1998). Thus, understanding the user's behaviour related to forming their goals and performing tasks is an important part of task analysis. As Norman (1988) says, "to get something done, you have to start with some notion of what is wanted - the goal that is to be achieved". Further, Norman (2013) provides us with a 7-stage model for analyzing the execution and evaluation of an action related to user's goals. In his model, the stages are the following:

1. Forming the goal
2. Planning the action
3. Specifying an action sequence
4. Performing the action sequence
5. Perceiving the state of the world
6. Interpreting the perception
7. Comparing the outcome with the goal.

Those which follow after forming the goal in the 7-stage model are the three stages of execution and evaluation. The three stages of execution (plan, specify, perform) and evaluation (perceive, interpret, compare) bridge the gap between the user's goal and all possible actions the user could take to reach that goal.

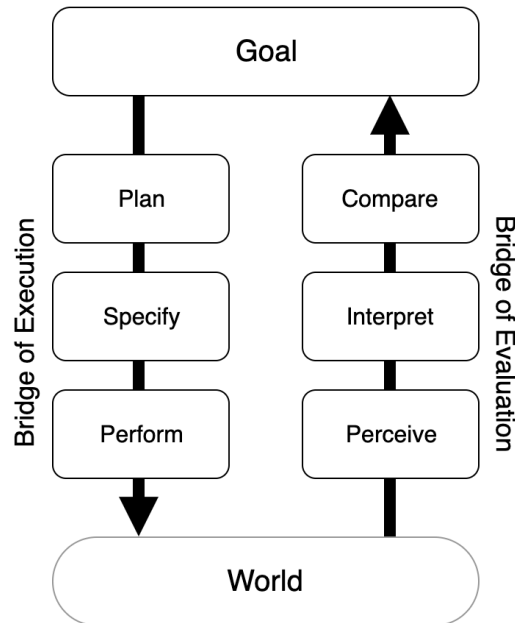


Figure 3: The seven stages of action model by Norman (2013).

Hackos and Redish (1998) provide an example of a user wanting to go outside through a door with the 7-stage analysis:

- | | |
|--|---|
| 1. User forms goal | Go outside from a room to get some fresh air |
| 2. User plans the action | Open the door |
| 3. User specifies action(s) | "It looks like I will pull this door handle here" |
| 4. User performs the action(s) | Pull on the door handle |
| 5. User perceives the state of the world | The door did not open |
| 6. User interprets the state of the world | "Well. That did not work. This door handle sure looks like I should pull it. I guess it does not mean that. I guess I need to push it" |
| 7. User compares the outcome with the goal | Did not get outside yet. (If the user still wants to meet the goal and still thinks this task is the best way to do it, the user goes through steps 3-7 again this time pushing on the door). |

Task analysis is relevant at all stages of the process. Courage et al. (2008) state that task analysis belongs everywhere in the process of planning, designing, developing, and evaluating a product. They continue, that task analysis should be done iteratively and that the focus, methods, granularity, and presentation of it may change over time as some aspects of it may become more or less relevant. Also, they provide questions to consider throughout a product development timeline, to assist the practitioner to gain a direction for task analysis at certain phases.

Table 12: Questions for task analysis at different phases of product development by Courage et al. (2008).

Stage	Examples of questions that task analysis should be used to answer
Strategic Planning	Why would someone or some organization choose to use this product? What goals in their world would this product help to meet? What benefits are most meaningful and valuable to users? How will this product be perceived by different cultures?
Predesign	What are the alternatives currently available and technologically possible that would address the why questions listed above? How do users achieve relevant goals today? What works well and what does not? How could our product make that easier?
Information Architecture	What do users know and what are their environments? How do users organize their world? What vocabulary do users use today for their goals and tasks? How can we incorporate that vocabulary?
Concept Design	What metaphors are familiar to users?
Interface Design	What do users know about interface conventions? How does the task flow of the new product match users' expectations from their current work? If we are changing users' task flows, how can we build in help for transitions?
Early Prototypes	What tasks should we provide to heuristic reviewers and usability testers?
Development	What does the user know that would address the problems we have uncovered? What changes should we make to the interface and information to better match users' expectations and work?
Post-release	How well does this release match the user/business needs that we uncovered in the original strategic planning phase? Are users now better able to achieve their goals than they were before they had this product?

Practical Reality Impinges on What We Actually Do. Courage et al. (2008) note that conducting task analysis, as any other practice in user-centred design, is dependent on several factors, which include:

- Time
- Resources
- People
- Availability of users to observe and talk
- Travel restrictions.

They highlight two possible issues that a design practitioner may face with proposing task analysis for a project: A difficulty facing market research and the perception of high costs on budget and time. Your internal or customer's product team may confront task analysis with claims that they have conducted sufficient user research and task analysis is not needed. (Courage et al., 2008) argue that market research tends to focus on breadth, whereas task analysis focuses on depth. For example, market research may find answers to which features users are interested in, where task analysis tells how users will use them. Furthermore, market research and task analysis may study different users (those who buy software vs those who use software).

Then, the perceived costs of task analysis: Courage et al. (2008) state the importance of time and resources for any product development and the hesitation of product teams to include activities that have an impact on these activities, if there is uncertainty on the value of the work. Results of the task analysis rely on the skills of the practitioner and each analysis include qualitative factors. More complex tasks may take more time to observe and require more expertise from the practitioner (Holzinger, 2005). But, Courage et al. (2008) argue for task analysis, that it saves money during design, development, and in the long run may turn into a profitable practice for the company. They say, that task analysis may cost lots of time and money upfront, but the benefits come from different stages of product development; It saves money during design when designers have a better grasp on how users work; It saves money during development when relevant features are recognized and unnecessary features can be left out. When a product meets the user needs, it can cut costs in customer support and training costs, overall reducing after sale costs for a product (Courage et al., 2008).

There are several types for task analysis. Hackos and Redish (1998) propose the following: Workflow analysis, Job analysis, Process analysis, and Procedural analysis. Ritter et al. (2014) propose the following: Hierarchical task analysis, Cognitive task analysis, GOMS (Goals, Operations, Methods, and Selection rules), and Keystroke level model (KLM). In this section, we will be delving deeper on Workflow analysis, Job analysis, and Hierarchical task analysis.

Workflow analysis is described by Hackos and Redish (1998) as a type of task analysis that focuses on a person's workflow. As simple as it may sound, a person's workflow

may contain dependencies on other people, or even have entry and exit conditions set by other people. In addition, there can be people in different locations and time zones, creating possibilities for more complexity for one's workflow. Hackos and Redish (1998) suggest conducting site visits with interviewing and observing people, to gain the best data for creating a solution that is attractive to the users. They emphasize that missing the holistic view of the workflow may result in designing a product that supports only a part of the it and may not support the rest of the workflow. In workflow analysis, the focus is to understand general tasks of person(s) in a workflow. The outcome of the analysis can be presented in a sequence diagram, to show how different tasks a linked together in a workflow and how the role might change between tasks. Example of a workflow analysis for a patient to get their medication from a pharmacy in Finland, including the roles of a patient, a pharmacist, a pharmacy clerk, a computer system, a fetching machine, and a ticker machine:

1. The patient visits their local pharmacy
2. A ticket machine handles matching the queueing patient to a free pharmacist
3. The pharmacist looks up the information from the patient's Kela card
4. The pharmacist asks for the amount of medication withdrawn from the prescription
5. The patient tells the wished amount of medication
6. The pharmacist inputs the information to their computer system
7. The computer system manages updating the information on the electronic prescription
8. A machine fetches the inputted amount of medication from storage
9. The pharmacist handles the receipt to the patient
10. The patient visits the pharmacy clerk and pays for the medication.

Job analysis is a type of task analysis that focuses on understanding what work a person does from their perspective (Hackos and Redish, 1998). In workflow analysis, work is analyzed on a horizontal level, understanding how a work can spawn between people. Then, Job analysis is about understanding on the vertical level how work flows through a single person, what different tasks are involved in a person's work day, week, or month. Job analysis can help the practitioner to find new marketing and development opportunities, as well as understanding importance of specific features that support the work of the person (Hackos and Redish, 1998). They suggest as the best practice to conduct Job analysis, is to spend time on the job with the person. Also known as "shadowing" or "walking in someone else's shoes", these practices provide more evidence than just interviewing the person, asking of their work from their manager, or reading their written job description. As the result of Job analysis, you want to list the tasks people do in their job, and consider the following information related to each task (Hackos and Redish, 1998):

- Frequency. How often do these people do each task?
- Criticality. How important is each task to their job?
- Time to complete. How time consuming is each task?
- Difficulty. How much of a problem is accomplishing this task?
- Division of responsibility. Do all the people in that job do this task? Do different people with the same job title do different tasks?

Hackos and Redish (1998) suggest combining both workflow and job analysis when conducting task analysis. The workflow analysis can help you find relevant user groups for product development, and job analysis helps in selecting relevant features.

Hierarchical task analysis (HTA), focuses on identifying, analyzing, and modelling user tasks within a system (Stanton, 2006; Zowghi and Coulin, 2005). Hornsby (2010) describes HTA as an analytical and descriptive tool that structurally and objectively describes users main goals and tasks , for creating an user understanding within the context of using a system. The goals and tasks can be defined to any desired level of detail (Annett, 2003).

For software design efforts, HTA helps designers to create User Interface and User Experience designs (Crystal and Ellington, 2004) that take into account identified user goals and tasks. HTA can provide understanding of the context of use through the analysis of users' goals and tasks in their environment. HTA can also answer the questions related to user requirements, when it is used to e.g. discover functional requirements of a system or a software through user tasks. As a result of HTA, one can produce a visual or written analysis and model of users' tasks and goals (Stanton, 2006).

Stanton (2006) proposes 9 principles to help practitioners conduct HTA.

Table 13: Stanton's (2006) principles 1-2 for HTA.

Principle	Explanation
1. Define the purpose of the analysis	HTA can be used to produce various data based on how a problem is approached with it. Therefore, defining whether HTA is used to specifying the use context of a software system, discovering better user understanding, or eliciting user requirements, will be important for producing wanted results.
2. Define the boundaries of the system description	According to Stanton (2006), the definition of system boundaries affects the target of HTA. Further, depending where the boundaries are drawn, inspection of tasks can happen on individual task level, a group of individuals' task level, or on whole system level, when inspecting the interaction between a system and a user.

Table 14: Stanton’s (2006) principles 3-7 for HTA.

Principle	Explanation
3. Try to access a variety of sources of information about the system to be analyzed	The reliability and validity of HTA should be assisted by usage of multiple data sources. Such sources can be observations, domain experts, interviews, various documentations (user manual, software documentation, guides, etc.) (Stanton, 2006).
4. Describe the system goals and sub-goals	Stanton (2006) states the overall aim for HTA is to create a sub-goal hierarchy for identified tasks. While HTA can be used to decompose high-level tasks into sub-tasks, Annett et al. (1971, in Stanton 2006) state that HTA aims to describe the identified sub-goals of the decomposed high-level task. Each new sub-goal is tied to an operation which is a sub-task of the high-level task (Stanton, 2006).
5. Try to keep the number of immediate sub-goals under any super-ordinate goal to a small number (i.e., between 3 and 10)	Stanton (2006) suggests to keep the amount of sub-goals from 3 to 10 for each super-ordinate goal. Further, he suggests to keep continual analysis of sub-goal groupings to check whether they make sense, and to also think about groupings of sub-goals if the groups exceed the limit of 10 sub-goals per a super-ordinate goal.
6. Link goals to sub-goals and describe the conditions under which sub-goals are triggered	Naturally, sub-goals should derive from the main goals. With complex tasks, sub-goals might relate to each other in more than just one way, thus it is important to consider relations between sub-goals and their trigger conditions (Stanton, 2006). Ormerod and Shepherd (2004, in Stanton 2006) proposes six different relations or sequences how sub-goals could interact with one another: linear (sequential), non-linear (non-sequential), simultaneous (concurrent), branching (choices), cyclical (repetitious), and selection (exclusive). Stanton (2006) emphasizes with relation analysis to also consider exit conditions to help formulate the hierarchical analysis and not get stuck in a loop between sub-tasks.
7. Stop re-describing the sub-goals when you judge the analysis is fit-for-purpose	Stanton (2006) suggests designers to consider when to stop the HTA analysis. Benefits for considering the stopping point include savings in time, costs, and effort. He suggests to think the stopping conditions for the analysis at the beginning, so that the scope of the analysis is well defined.

Table 15: Stanton's (2006) principles 8-9 for HTA.

Principle	Explanation
8. Try to verify the analysis with domain experts	Annett (2004, in Stanton 2006) suggests to verify the HTA analysis. Stanton (2006) continues with the benefits of "verification of the completeness of the analysis and help the experts develop a sense of ownership of the analysis". Modelling the analysis into a visual and orderly model can help in explaining the analysis to domain experts and relevant stakeholders.
9. Be prepared to revise the analysis	With complex tasks, it might be necessary to go over the HTA analysis through 10 iterations, if there is time available, as a first pass analysis is not enough to produce sufficient details (Stanton, 2006). It is important to do multiple revisions to ensure that the analysis is broad enough and takes into account necessary actors in the system, their sub-goals and sub-tasks.

Vredenburg et al. (2002) studied commonly used user-centred design practices in the industry. Their survey states that task analysis helps in understanding the context and validity of results produced by other design practices. However, conducting a task analysis was considered slow, as a formal task analysis takes a considerable amount of time.

3.1.3 User stories

User stories are short sentences often used in agile software development. Those contain the essential information to tell the reader the following: who the story is addressing, what goal is being pursued, and optionally, what is the perceived benefit or reason for achieving the goal (Cohn, 2004; Lucassen et al., 2016). Cohn (2004) recommends user stories to be used with Agile methods. Such methods include e.g. Scrum or Kanban. In Scrum, user stories can be used as Product Backlog Items that describe the different parts of the project or a certain requirement to be fulfilled (Deemer et al., 2012). Collections of such Product Backlog Items form Epics, which are higher level goals that all user stories and other artefacts linked to it try to help achieve.

User stories follow a format. Cohn (2004) presents a simple template that can help in forming a user story: "I as a (role) want (function) so that (business value)". In his template, there are three parts in parenthesis: role, function, and business value. Lucassen et al. (2016) propose a conceptual model for user stories, where they have disseminated them to three parts, similarly as Cohn (2004) has in his template. The parts they propose: the role, the means, and the end of a user story.

The role defines who is in question in the user story. In other words, the single relevant actor, which the story addresses (Cohn, 2004; Lucassen et al., 2016). Identified user segments or stakeholders are a good resource when looking for relevant roles in user stories. In addition, personas or fictive users can be used if necessary (Lucassen et al., 2016) e.g. in the early development/design phase.

The means of the user story describe the goal or task that the user is trying to achieve or complete. Lucassen et al. (2016) propose three grammatical approaches to writing the means of a user story by considering *a subject*, *an action verb*, or *a direct object*. A subject with an aim or need, such as "want" or "am able", describes the link between the user and the form of approach towards the wanted goal or task. An action verb, such as "feel" or "browse", further elaborates the type of action that is taken in the form of approach. Further, Cohn (2004) suggests in his book to write the user stories in active voice. And finally, a direct object is needed to link the subject's aim and action to the user's goal or task.

The end of the user story in Lucassen et al.'s (2016) paper describes similar aspects as the business value in Cohn's (2004) template. In both cases, the end of the user story, which comes after the goal or task description, gives additional information to the reader about the reasoning of the user story. Cohn (2004) suggests in his template to describe a business value, which can be interpreted as the motivation or feelings the user has or experiences while undergoing the situation described in the user story. Lucassen et al. (2016) add in their paper three additional descriptive elements to the end of the user story:

1. Clarification of the means. Above was mentioned the business value as an example of a clarification to the means. Writing the clarification does not have to follow any particular form, as long as it delivers the message clearly, precisely, and uses minimal amount of words as possible. Example by Lucassen et al. (2016): "As a User, I want to edit a record, so that I can correct any mistakes".
2. Description of a dependency on another functionality. The end of a user story can give hints about a dependency to another functionality in the designed software. E.g. a user story such as "As a User, I want to open the interactive map, so that I can see the location of landmarks" suggests that the interactive map is tied to a landmark database that has to exist for this user story to be possible (Lucassen et al., 2016).
3. Description of a quality requirement. In the end of the user story, we can describe quality requirements that are connected to a particular task or action in the user story. E.g. "As a User, I want to sort the results, so that I can more easily review the results" promotes easy of use (Lucassen et al., 2016).

Lucassen et al. (2016) emphasize that these descriptive elements are not mutually exclusive and all three may appear in the same user story. When writing user stories, all descriptive elements are worth noting so that one can make the most out of user stories. However, the quality of the user stories can be violated despite conforming

to the descriptive elements. For example, description of a dependency on another functionality usually requires the user story to be split into individual stories. In the example user story "As a User, I want to open the interactive map, so that I can see the location of landmarks" should be split into two separate stories, which the authors give the following suggestion: (1) "As a User, I want to open the interactive map", and (2) "As a User, I want to see the location of landmarks on the interactive map".

The quality of a user story can be measured according to Lucassen et al. (2016). They propose a framework for Quality User Stories (QUS). They studied characteristics of requirements and agile requirements quality, and constructed a model for categorizing the important quality criteria of a user story.

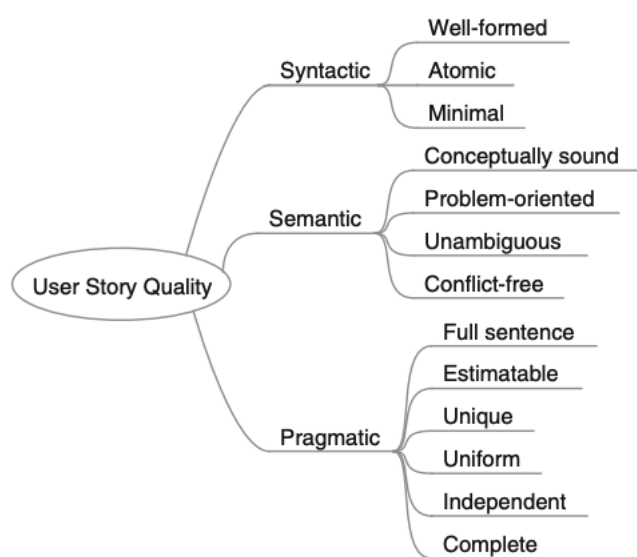


Figure 4: Model for important quality characteristics of a user story (Lucassen et al., 2016).

They categorize the 13 different quality criteria under three main categories: Syntactic, Semantic, and Pragmatic. These three main categories are Lindland's (1994, in Lucassen et al. 2016) categories, which is described by Lucassen et al. (2016) as follows: The syntactic quality concerns the textual structure of the user story. The semantic quality concerns the relations and meanings of the story content. And lastly, the pragmatic quality considers the aspects of subjective interpretation of the user stories. In the following tables, the qualities are further elaborated according to their work, giving a description of the syntactic, semantic, and pragmatic quality criteria. Then, Table 18 provides examples for user stories that violate any User Story Quality criteria.

Table 16: Syntactic and semantic user story quality criteria by Lucassen et al. (2016). "Ind" means individual.

Criteria	Description	Ind/Set
<i>Syntactic</i>		
Well-formed	A user story includes at least a role and a means	Ind
Atomic	A user story expresses a requirement for exactly one feature	Ind
Minimal	A user story contains nothing more than role, means, and ends	Ind
<i>Semantic</i>		
Conceptually sound	The means expresses a feature and the ends expresses a rationale	Ind
Problem-oriented	A user story only specifies the problem, not the solution to it	Ind
Unambiguous	A user story avoids terms or abstractions that lead to multiple interpretations	Ind
Conflict-free	A user story should not be inconsistent with any other user story	Set

Table 17: Pragmatic user story quality criteria by Lucassen et al. (2016). "Ind" means individual.

Criteria	Description	Ind/Set
<i>Pragmatic</i>		
Full sentence	A user story is a well-formed full sentence	Ind
Estimatable	A story does not denote a coarse-grained requirement that is difficult to plan and prioritize	Ind
Unique	Every user story is unique, duplicates are avoided	Set
Uniform	All user stories in a specification employ the same template	Set
Independent	The user story is self-contained and has no inherent dependencies on other stories	Set
Complete	Implementing a set of user stories creates a feature-complete application, no steps are missing	Set

Table 18: List of example user stories that violate user story quality criteria (Lucassen et al., 2016).

ID	Description	Violated qualities
US ₁	I want to see an error when I cannot see recommendations after I upload an article	<i>Well-formed</i> the role is missing
US ₂	As a User, I am able to click a particular location from the map and thereby perform a search of landmarks associated with that latitude longitude combination	<i>Atomic</i> two stories in one
US ₃	As a care professional, I want to see the registered hours of this week (split into products and activities). See: Mockup from Alice NOTE—first create the overview screen—then add validations	<i>Minimal</i> there is an additional note about the mockup
US ₄	As a User, I want to open the interactive map, so that I can see the location of landmarks	<i>Conceptually sound</i> the end is a reference to another story.
US ₅	As a care professional I want to save a reimbursement—add save button on top right (never grayed out)	<i>Problem-oriented</i> hints at the solution
US ₆	As a User, I am able to edit the content that I added to a person’s profile page	<i>Unambiguous</i> what is content?
US ₇	As a User, I am able to edit any landmark	<i>Conflict-free</i> US ₇ refers to any landmark, while US ₈ only to those that user has added
US ₈	As a User, I am able to delete only the landmarks that I added	See above text
US ₉	Server configuration	<i>Well-formed, full sentence</i>
US ₁₀	As a care professional I want to see my route list for next/future days, so that I can prepare myself (for example I can see at what time I should start traveling)	<i>Estimatable</i> it is unclear what see my route list implies.
EP _A	As a Visitor, I am able to see a list of news items, so that I stay up to date	<i>Unique</i> the same requirement is both in epic EP _A and in story US ₁₁
US ₁₁	As a Visitor, I am able to see a list of news items, so that I stay up to date	See above text
US ₁₂	As an Administrator, I receive an email notification when a new user is registered	<i>Uniform</i> deviates from the template, no "wish" in the means
US ₁₃	As an Administrator, I am able to add a new person to the database	<i>Independent</i> viewing relies on first adding a person to the database
US ₁₄	As a Visitor, I am able to view a person’s profile	See above text. This feature is dependent on having a person created before viewing.

3.1.4 Scenarios

Scenarios are stories placed in the real world that depict how one user or many users would approach or use a system within a certain environment with certain tasks and goals (Carroll and Rosson, 1990; Carroll, 1999). One could think of scenarios as extended user stories. In addition, Alexander and Maiden (2004) highlight the scenario stories to be readily understood, told in a way non-technical people can understand, and errors and omissions can be spotted in the story. In essence, the scenarios have to speak the user's language. Scenarios can help designers to explain user tasks and goals within a system through a written story, and these can be used as a foundation for other tools, such as use-case diagrams, task analysis, or creation of prototypes.

Scenarios have multiple use in design. They can be used in requirements analysis, development, definition processes, and general design practices, as preparation material for prototyping, documentation material, and as supporting document to other practices (Carroll, 2002). In this section, scenarios are treated as design tools, helping in creating solutions to users' problems.

The Interaction Design Foundation suggest few steps on how to create scenarios (Interaction-design.org, 2019):

1. Gather a design team of relevant stakeholders and a design space to conduct a workshop between 2 to 3 hours.
2. Prepare post-it notes, flip charts or any other tools that can help your team to write their scenarios in such a way that they are visible and accessible for everyone in the team to read.
3. Make sure everyone in the design team understands the objective of the sessions and the core concept of scenarios to get designers on the same page.
4. Provide ground information of the context and users if possible. Any information that helps designers understand the setting, context, and users will be beneficial for creating scenarios.
5. When writing the scenarios, it is good to focus on the context and the following questions: who, what, when, where, and why? These questions help create depth and detail to the scenarios. Also, try to write them from the relevant actor's (a user or a group of users) perspective, and include any other actors as necessary. In relation to this, Carroll (1999) suggests scenarios to be accessible to various stakeholders, meaning that the scenarios speak user's language and does not shun away non-technical people. Also, scenarios can be used as a communication tool to people outside of the project team Bødker (1999).
6. Write the scenarios on the tools you had prepared for the workshop for accessible reading and visualization of scenarios. Then, be prepared to evaluate the scenarios by going through them within the team and then group them for better sense-making and spotting of gaps and errors in the stories.

7. Repeat the process as many times as needed to map the key tasks and goals of the users.

Scenarios have different depths in detail when describing user's actions and goals. They can be very detailed in their description, depending on how far the designer is willing to use time and effort to refine the scenarios. They can be constructed with small steps and start with vague details to get the direction right. Scenarios can be very rough initially, that do not go into full detail how a task is carried out or how the system allows these tasks to be carried out (Carroll, 1999). On the other side of the spectrum, scenarios can be very detailed in their description of system use (Carroll, 2002). Within the domain of scenario-based design, that means scenarios can be used with a 'solution-first' strategy (Cross, 2001). This method is about forming the solution to a problem without knowing or having properly defined the real problem, making the method sound paradoxical (Cross, 2001; Carroll, 2002).

While scenarios can differ between depth, they can have multiple views as well. Scenarios can inspect the moment-to-moment thoughts, emotions, and experiences of the users for a more cognitive view, or it can inspect the moment-to-moment actions for a more functional view (Carroll, 1999). They are not exclusive in a way that a scenario should provide either view, instead, a scenario can contain multiple views and they may enhance the content and depth of the story.

Such as interviews, scenarios can be open-ended or closed by their descriptive nature (Bødker, 1999). According to her, open-ended scenarios are useful in the early design phases of an iterative, user-centred design process. In one of her case study projects, open-ended scenarios promoted designers to think of future steps of the design to fill in the blanks for solution in the scenarios. Further, with closed scenarios, she suggests looking for opportunities in the design process such as using the closed scenarios as a backbone for testing or prototyping. In addition, her case studies show promise for iterative user-centred design, where scenarios provide feedback for designers and opportunities in utilizing them in early or later design phases. Further, to summarize scenarios, they ".. are not just a detached description of user tasks and actions but selective scripts or stories that stage user actions .." (Bødker, 1999).

With scenarios, one can utilize a practice called scenario-based design. As defined by Rosson and Carroll (2008) and Carroll (2002), "scenario-based design is a family of techniques in which the use of a future system is concretely described at an early point in the development process". For this, descriptive narratives of actions, users, and goals, such as scenarios themselves play a part in scenario-based design. In scenario-based design, the focus is in creating early solutions to an ill-structured problem (Rosson and Carroll, 2008). Reitman (1965, in Carroll 2002) defines the ill-structured problem to have the following three characteristics: "the problem state is not fully specified", "possible moves in the problem space are not all given", and "the goal state is not specified".

Creating solution without properly knowing the problem domain can be risky. The created solutions or partial solutions can be sub-optimal or even on an incorrect path (Carroll, 2002). Further, fixation on such solutions causes designers to stray further away from the ideal solution. There is a risk of staying fixated to an attractive solution. A general risk related to a written specification of users, if users are not involved in creation of such specification, designers may inject misinterpretations of their users, leading to more misunderstandings in both design efforts (Carroll, 2002; Rosson and Carroll, 2008), and user understanding. With many solutions at hand, designers may easily drown in the sea of documentation. Documenting each scenario, keeping them up-to-date as the product or service moves on with the development efforts and possibly changing requirements, can cause lots of effort needed to maintain the scenarios relevant to the project (Rosson and Carroll, 2008). Another risk is with existing scenarios, designers may focus much on something existing and attempting to develop those further instead of looking for other solutions to expand the design and solution space (Rosson and Carroll, 2008). But, not all risks are necessary a hindrance to the project, as Rosson and Carroll (2008) mention that designers need constraints to keep the amount of ideas and solutions in check.

Despite the negatives, scenario-based design can prove fruitful design mechanism that help overall design efforts. Creating many rapid solutions can help expanding the design space and definition of the problem state by provoking questions, analogies, and connections related to them (Carroll, 2002). Scenarios also help designers in reflecting their design decisions (Rosson and Carroll, 2008). They say, by reflecting one's decisions, it is about thinking of their own competence; the strength and weak points of their design.

Scenarios can be work-oriented and describe how users handle the use of a system (Carroll and Rosson, 1990). Realistic scenarios can provide proper insight on user requirements, needs, concerns, tasks, goals, and interactions with their environment (Rosson and Carroll, 2008). They say, that scenarios are made effective when the users are invited in creating them together with the designers. In other words, scenarios can be seen as a medium to gather designers and users in creative meetings and address the usefulness of a system (Bardram, 2000). Rosson and Carroll (2008) continue, while scenarios are a great design tool, they are not a remedy to design challenges. But used in conjunction with other design practices, scenarios can prove useful in design efforts.

3.1.5 Prototyping

Prototypes are tools that allow a designer express their vision for a product, a developer to assert the feasibility of visual decisions, a user to experience the feel and the look of a prototype, and in general, prototypes are design items that help the products in development reach a desired state in its life cycle.

Beaudouin-Lafon and Mackay (2008) describe prototypes as concrete representations of an interactive system, either a part or all of it. As such, the prototype cannot

be an abstract description of a system, but a tangible, interaction allowing artifact for relevant shareholders to vision the end result of a product, they continue. In essence, prototypes give body to an idea for a visual presentation (Buchenau and Suri, 2000; McElroy, 2016), that can be used for communicating your idea to others or for testing purposes with the designed user groups, to improve the idea of the prototype further (McElroy, 2016). In summary, "prototypes are a representation of all or part of an interactive system, that, although limited in some way, can be used for analysis, design, and evaluation" (SFS-EN ISO 9241-210, 2019).

When creating prototypes, there are several aspects to consider: Which questions are we trying to answer with the prototype? Are we looking to improve something existing? Are the created prototypes how significant in relation to the project at its current state? Prototypes can answer various questions depending on the usage. For example, prototypes can be used to assess feasibility of a design, assess the usability of a design, prioritization of features in software, further exploration of a problem, creation of alternative solutions, and many more (Beaudouin-Lafon and Mackay, 2008; Warfell, 2009; McElroy, 2016). It would be extremely lengthy to attempt list all possible solutions prototyping could offer, thus every project should evaluate at its state the most important questions it needs answered, and whether prototyping can offer a solution to it. But, to continue the aforementioned list, McElroy (2016) proposes few reasons why we should create prototypes.

Understanding the problem context. McElroy (2016) states, that prototyping allows designers to understand the problem they are solving and possibly discover alternative problems related to the main problem. Finding the root-causes of the problem at hand helps in understanding the needs of your users, as well as realizing which problems need solving the most and how can your solution serve its users the best. Finding the main problems to solve allows the project to find the desired direction early and allows for cost savings from making changes later in the project (McElroy, 2016). Not only does prototyping create cost savings, according to Warfell (2009), prototyping:

1. allows the right people (designers and developers) to make right decisions for the prototype,
2. allows the best prototype to emerge from many other,
3. is adaptive and quickly updated,
4. reduces misinterpretations between parties with a tangible medium to support conversations about the product,
5. brings focus to the project,
6. highlights mistakes early on with the prototype, reducing rework costs by avoiding making errors,
7. and reduces overall risks associated with the project by reducing misinterpretations and finding problems during the product development process.

Finding alternative solutions with prototypes. Creating multiple design ideas and concepts, and turning them into prototypes will eventually lead to finding alternative solutions. McElroy (2016) suggests to design multiple solutions to users' needs to create breadth to solutions. This allows designers to evaluate and test different approaches to pick the best ones, and helps in avoiding getting stuck on a single promising solution. Exploring alternate solutions and having real users test them leads to more confident design choices (McElroy, 2016).

Understanding product value. Prototyping can be used to review different aspects of a product: the perceived product value for the organization and the users, and understanding how users will behave with the product. The perceived product value for an organization can be evaluated by assessing the business strategy behind the product development (McElroy, 2016). Tools, such as a business model canvas or lean canvas, provide the big picture of a product's goals it attempts to achieve on a long run, which can be also called the roadmap of the product. Roadmapping benefits from a wide source of inputs that help crystallize the vision and goals for a product. Prototyping is one such source of input, that gathers visual feedback and experience feedback from organization stakeholders and end users.

Understanding user flows. With modern tools such as Adobe XD, designs and low-fidelity prototypes can be produced much more easily than before. But, with speed there is a higher risk of omitting users in design and focusing on a list of requirements or assumptions of users. However, when we combine the speed of creating designs, turn them into early prototypes and involve real users in test scenarios, we can learn more how well our designed prototype meet the users' needs. Prototyping helps in understanding the user flows and requirements in design and development to support those flows, and then understanding better the UX and UI aspects and elements of the prototype and how they affect the users (McElroy, 2016).

A prototype can act as a tool for communication. Designers and developers may find themselves often in the need to communicate their ideas to relevant stakeholders. Without a prototype acting as a medium to visualize the idea of the end product, conversations about the design may be misinterpreted. Humans have unique mental models to help them visualize different ideas (McElroy, 2016), and even when someone is nodding in agreement to a proposal, it does not ensure that both you and the other person understand the idea in the same way. So, to avoid misguided expectations and misinterpretations, having a prototype available during conversations will help in linking the concept, ideas, and talks to the prototype, and keep conversations focused around it if necessary (McElroy, 2016).

Agile prototyping brings in an iterative and evolving activity that is supported by a good process (Warfell, 2009). In software projects, each project ordered by the customer is often unique in nature, which puts pressure on design processes. The same process may not apply in the same way to all projects, thus processes must be adapted to each project to provide the best support (Warfell, 2009). So what is a good process? Good processes are discovered through trial-and-error, from case

studies, and learning from best practitioners. The following agile prototyping process is proposed by Warfell (2009) and by no means is it the perfect process, but, it can serve as a starter and every design practitioners should be ready to modify their processes to suit their project needs and doing so maximize the benefits.

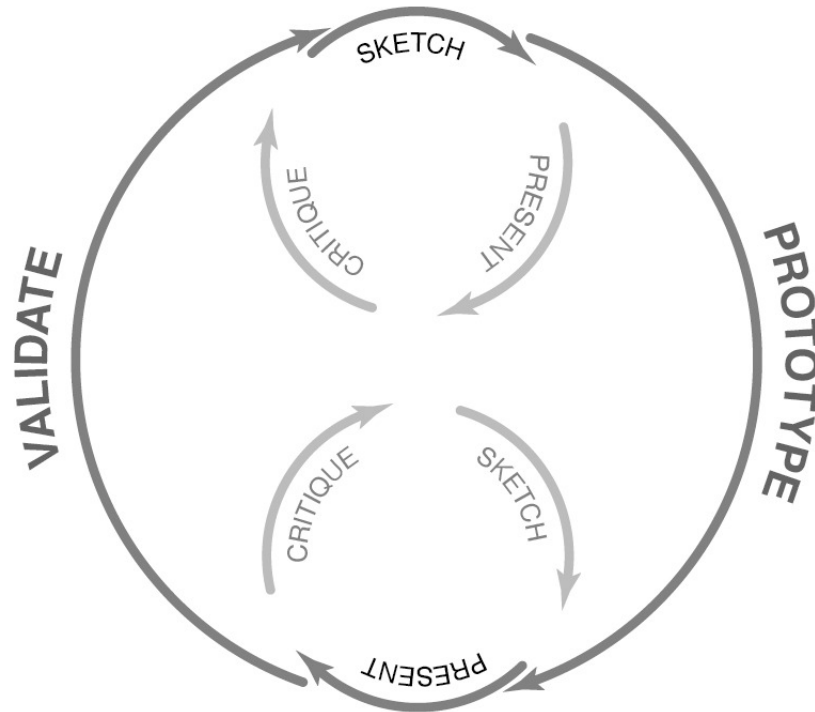


Figure 5: Iterative prototyping process by Warfell (2009).

The prototyping process by Warfell (2009) contains four major steps: Sketching, Presenting & Critique, Modelling (prototyping), and Testing (validating). This is visualized in above figure.

The process starts from the top with sketching. Then, it moves on to presenting and giving critique to the sketch before it can be modelled and presented. Once modelled, it will be presented. Then it receives critique for improvements. The modelled prototype is then tested as necessary. This concludes an iteration and the process continues iteratively, with new or existing parts being sketched, presented, critiqued, modelled, presented again, critiqued, re-sketched, and tested. The following four paragraphs describe each of the major steps in the process:

Sketching. In sketching, the aim is to produce as many ideas as possible and to turn them into low-fidelity images. These images should have enough detail to convey the designer's idea to the person inspecting the sketch (Warfell, 2009). Different tools can be used to sketch out an idea: paper sketches, white board sketches, sketching software such as Adobe XD, UXPin, Sketch, PowerPoint, or creating fast low-fidelity prototypes by coding. Many frameworks and libraries online allow for fast prototyping by coding. One should be cautious with creating software

prototypes, as customers and other stakeholders may not fully understand what a software prototype means. They can easily suggest making the software prototype into a functioning end product without sanitizing or refactoring the pieces of code that were appropriate for a prototype, but not for an end product.

Presentation and critique. After sketching the ideas, it is time to move on to presenting them and giving critique. When presenting and giving critique, it is a good practice to set a time limit. Warfell (2009) suggests to keep presenting to 3 minutes and giving critique to 2 minutes. The process of evaluating each sketch by presenting and giving critique, is to find the best ideas amongst many and the more people and from different backgrounds we have present during the critique phase, the more richer feedback can be acquired for each of the presented sketches. Further, if the sketches are done on a paper or a white board, it is a good practice to write notes from critiques directly onto the paper (Warfell, 2009), and in the case of software sketches, into the software itself or as comments in code.

Modelling. Presenting and giving critique leaves us with the best ideas which can then be taken into a prototype. When the design is being modelled into a prototype, there are several consideration to be taken into account, according to Warfell (2009):

- What modelling tools are being used to create the interactive prototype?
- For which audience or user segments is the modelled prototype designed for? If a prototype addresses only a certain sub-segment, this should be taken into consideration when planning tests with the prototype to have the correct user group representatives join the testing session.
- Time available to model the prototype. Managing the time and costs for creating prototypes is one key concern for prototyping (Camburn et al., 2017).
- Level of fidelity. Prototypes provide different feedback from users depending on the fidelity, as a lower fidelity prototype might not provide detailed feedback on some UI elements or the feel of the prototype as much as a higher fidelity one. Further, the fidelity of the prototype can affect whether it provides no feedback (too low fidelity, done too fast) or provides good feedback, but has too much detail (too high fidelity, done for too long) (McElroy, 2016). Finding the just right amount of time and effort invested to modelling a prototype is a challenge (see Figure 6).

Testing. Testing the prototype can be done in several ways, and in this section we talk about testing internally, with the client, and with the users. When planning prototypes, it is important to test them internally with the development and design team before testing with the client or the users to avoid any embarrassing errors on the parts of the prototype that should be testable. Testing with the clients is a form of acceptance testing, where agreed features for the product are fleshed out in the prototype. The test session should be collaborative, such as the sketching, presenting, and critiquing phases in the process, to allow the client to have the means of affecting

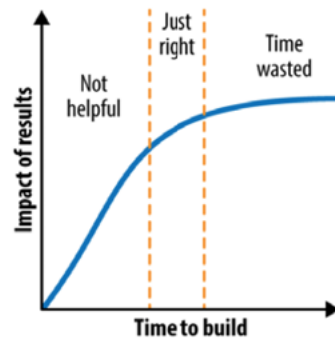


Figure 6: Too low fidelity prototype may not be useful, but a too high fidelity prototype may result in wasted time (McElroy, 2016). Finding the 'just right' fidelity within the satisfied time to build can be difficult.

the prototype and ultimately being part of the developing process (Warfell, 2009). A test with the client is a good way to provide evidence for functioning or non-functioning ideas and that way find the best design choices for the end product by modifying the prototype as many time needed within the scope of time, budget, and resources. Testing with the users provide feedback on user flows and whether your prototype truly supports the users' processes and needs. These can be discovered and verified through usability tests with enough participants to provide an adequate amount of qualitative data to create assessments of the prototype (Warfell, 2009).

3.1.6 Evaluation of design choices

Evaluating design choices and results are required in user-centred design activities (SFS-EN ISO 9241-210, 2019). Design practices help us to create an analysis and representation of information regarding the relationship of users' and systems in various perspectives, depending on the practices used. Thus, to validate, discover new, or understand better, evaluation is needed in design (SFS-EN ISO 9241-210, 2019). Designers should assess, what kind of and how evaluation fits in the design process either within their organization on the general level or tailored to meet the requirements and needs of a single project.

Different approaches exist to evaluation of design. One common approach is to conduct usability tests to evaluate the usability of the system, as a form of user-based testing (SFS-EN ISO 9241-210, 2019), to find answers to different aspects of usability, such as effectiveness, efficiency, or satisfaction with the system (SFS-EN ISO 9241-11, 2018). Further aspects for usability, according to Nielsen (1993, p. 26) and Quesenbery (2001), include minimizing errors, and supporting learnability and memorability. Another common approach is to use inspection-based evaluation practices based on usability and accessibility guidelines or requirements (SFS-EN ISO 9241-210, 2019). These practices can be heuristic evaluations, cognitive walkthrough, or expert analyses against said criteria. While they provide fast, cost-effective,

and broad analysis, inspection-based evaluation may not be able to provide the same depth in analysis and results as user-based testing (SFS-EN ISO 9241-210, 2019).

The results and process of evaluation is affected by the number of users involved in the evaluation. Common sense and quantitative research suggests that higher amount of data points provides more accurate data. Often in design projects, time and money are common constraints that drive designers to think for the fastest and sufficient enough solutions. The sufficient amount of users for design evaluation may vary depending on scope and other constraints, but in general, an amount between 5 and 20 may contribute towards sufficient results (Hackos and Redish suggest 5 to 10, Beyer and Holtzblatt 6 to 20) (Beyer and Holtzblatt, 1998; Hackos and Redish, 1998). On the other hand, Wixon (2003) argues that considerations towards the goal and quality of the evaluation methods are more important than asking "how many users" are required for evaluation and analysis. Different methods for evaluation and the criteria to select the most appropriate can be found from the technical report ISO/TR 16982 (Bevan, 2009).

In the field of usability engineering, design evaluations are often classified into two classes: formative and summative evaluation (Faulkner, 2000, pp. 138-139). Formative evaluation is about supporting the design process by refining and evolving the design based on internal or user feedback from testing of the design with appropriate task scenarios (Gabbard et al., 1999; Faulkner, 2000; Ahmed et al., 2006). Summative evaluation is done to a complete part of a system, with the aim of defining usability metrics and giving measurements against them (Faulkner, 2000, p.138-139). Gabbard et al. (1999) and Ahmed et al. (2006) describe a summative comparative evaluation, where the complete design's usability in interaction design is measured against other competitive systems. In carrying the evaluation, Faulkner (2000, p. 140) suggests to take note on identifying the correct target group for evaluation and to recruit users who represent a proper user-type with the necessary range of skills. Furthermore, she suggests to recruit more than the perceived required amount of users, in case not everyone is able to attend and having reserve users at hand.

The design practices described in this thesis have use in the evaluation of design. Interviews are great for finding the things that work and do not work in a particular design. Asking users directly on their feelings and reactions towards a design of a current or an envisioned future product can yield detailed evaluations from the users' perspective. If, for example, users are not good at verbally elaborating their opinions on the design, another approach to evaluating a current or future design is to allow the users to interact with it. Understanding how the users approach, interact, and interpret the product or system design through task analysis is an effective alternative to an interview. User stories are similar to shortened transcripts of an interview, where the user's role, tasks, and goals are written. Those stories can translate into user requirements, which will guide the software development work and set constraints for the design. When evaluating the design against user requirements, re-creating the user stories with users or re-visiting old stories can be

used for evaluation. Scenarios work in the same fashion as user stories, although their content contains more descriptive elements of the user, her tasks and goals. Prototypes provide a user-testable medium to evaluate the proposed design for the product. Prototypes offer possibilities for usability evaluation and usage of other design practices alongside the prototype, to create focus for evaluation activities to revolve around the prototype's design.

3.2 User understanding

User understanding, in this thesis, is defined through SFS-EN ISO 9241-11 (2018) and SFS-EN ISO 9241-210 (2019) definition of specifying the context of use, its elements, and the user requirements. SFS-EN ISO 9241-11 (2018) defines the context of use through four elements: the users, the goals and tasks, the resources, and the environment.

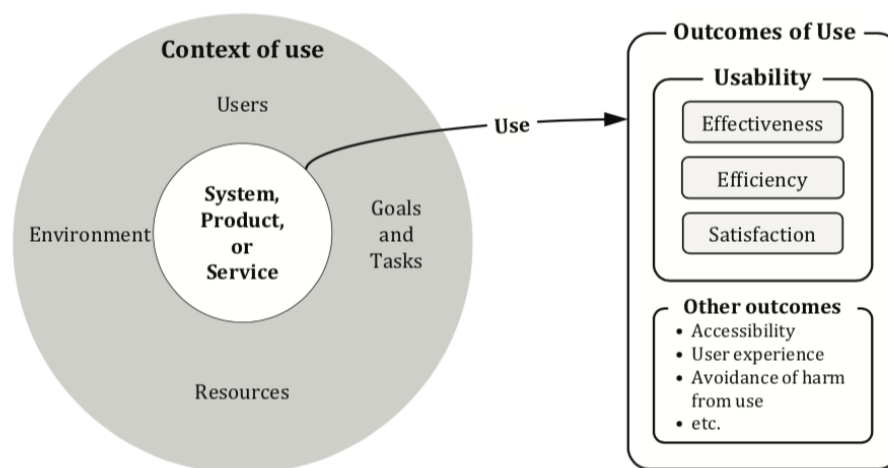


Figure 7: SFS-EN ISO 9241-11 (2018) definition of usability that results from the use of a system, product or service in a context of use.

Emotional motivations and values are also inspected as part of the user understanding definition. While user understanding is a large definition itself, the provided perspective should allow for a meaningful inspection for evaluating a software product or service.

SFS-EN ISO 9241-210 (2019) highlights the importance of understanding the context of use and the specification of user requirements as part of human-centred design. Understanding the context of use, in the object of inspection (e.g. a product or a service), assists in eliciting the needs, difficulties, shortcomings, and constraints for system improvement (SFS-EN ISO 9241-210, 2019). Specifying the user requirements is the next activity in human-centred design process, where understanding the user and stakeholder needs according to the specified context of use, allows for planning quality solutions and lay the foundation for design work (SFS-EN ISO 9241-210, 2019).

The human-centred design process defines an iterative design work process that is based on "explicit understanding of the users, tasks, and environment" and where the users are involved throughout the process (SFS-EN ISO 9241-210, 2019).

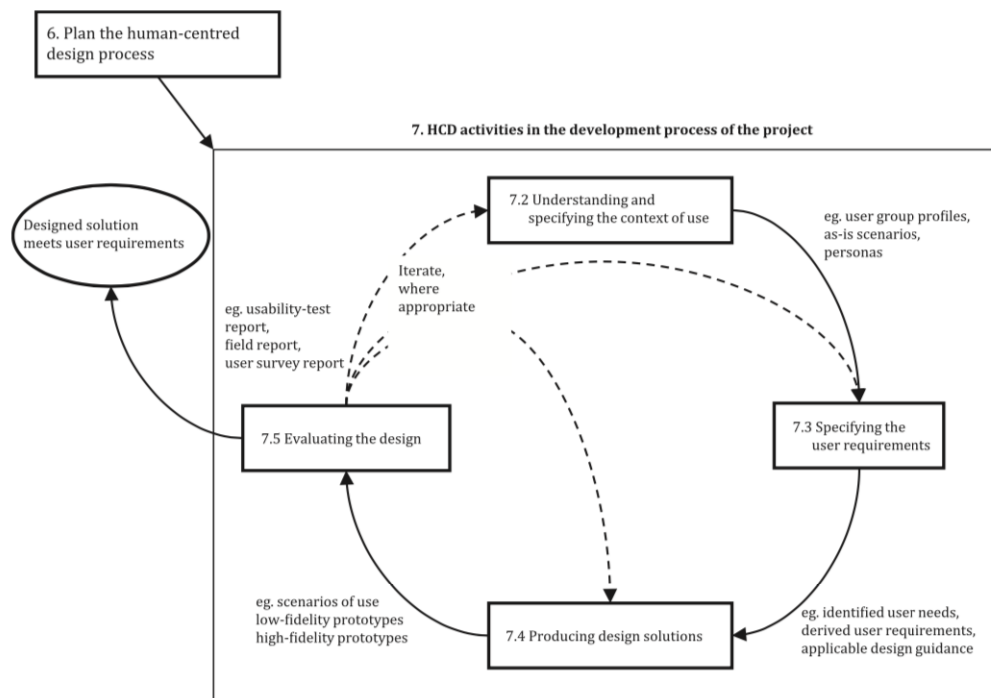


Figure 8: SFS-EN ISO 9241-210 (2019) definition of human-centred design activity.

One should consider the evaluation of emotional inhibitors and motivators in specifying user understanding. Cerejo (2018) notes that in addition to specifying rational aspects such as user goals, needs, and expectations, one should focus on emotional aspects as well. Rational decisions may not tell of the subconscious motivators and inhibitors of the rationale action.

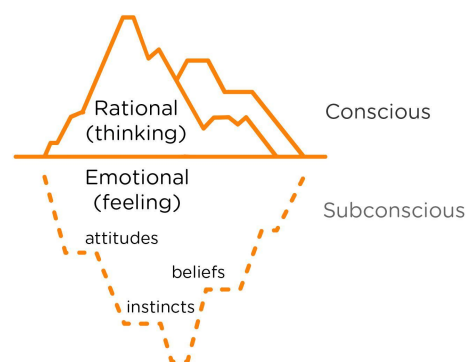


Figure 9: Subconscious emotional drivers may inhibit or motivate conscious actions, according to Cerejo (2018).

To help defining user understanding, we can ask the questions of who, why, and how an object of inspection (product, service, or system) is used. **Who uses?** It is important to define the users of the product or service, in order to understand who are the users or stakeholders of the object of inspection. In the SFS-EN ISO 9241-11 and SFS-EN ISO 9241-210 standards, they define the importance of understanding and specifying the context of use, and further in the context of use, the users (SFS-EN ISO 9241-11, 2018; SFS-EN ISO 9241-210, 2019). In addition to knowing and understanding the users, their environment, goals and tasks, and resources available provide information that can contribute towards segmenting users and understanding the limitations and requirements for users' use context (Courage et al., 2008; SFS-EN ISO 9241-11, 2018). Understanding the user segments helps in categorizing the results of why and how the users use the object of inspection. In addition, some segments can give hints on functional user requirements. For example, users with limited or weak eyesight can benefit more from interactive software systems with haptic feedback rather than visual feedback.

Why they use? For achieving a goal or conducting a tasks, motivation is involved (Gautam, 2012). As human, our motivation affects our actions, and the outcomes then produce emotions as feedback. Cerejo (2018) highlights the importance of not just understanding the observable actions, tasks, and goals of a user, but to also understand the emotions related to those actions, tasks, and goals. The emotions a user experiences from the use of the object of inspection may positively or negatively affect the motivation of the user to continue using it. Understanding the positives and negatives of UX contributes towards customer value analysis, which can be used e.g. to survey the perceived benefits and sacrifices to summarize the value(s) the customer receives in the context of get and use of a product or service (Kauppinen et al., 2009).

Inspecting the emotional inhibitors and motivators could be looked through the evaluation of perceived gain or loss of value. Defining the perceived gain or loss of value is simple: a user is asked whether they see some value in the object of inspection or that it is missing some value that another object provides. Almquist et al. (2016) provide a value pyramid that consist of 30 distinct elements of value for evaluating the perceived value of a product or service from the consumer perspective. According to them, the values and their categorization takes after Maslow's "hierarchy of needs". Maslow's (1943) model contains five basic needs (physiological needs, safety needs, love needs, esteem needs, and self-actualization needs, see Figure 11).

In Almquist et al.'s (2016) model, the value categorization follows Maslow's concept but turns it into a modern heuristic model of values, which reside in four categories: functional, emotional, life changing, and social impacting values. In assessing the values a product or a service provides to a user, and to inspect motivating or inhibiting factors, we can use their model as the model for defining different values a user may hold towards a product or service. Furthermore, it should be noted, that value emerges for users when they make use of a product or a system, and it truly provides solutions for the user (Kauppinen et al., 2009).

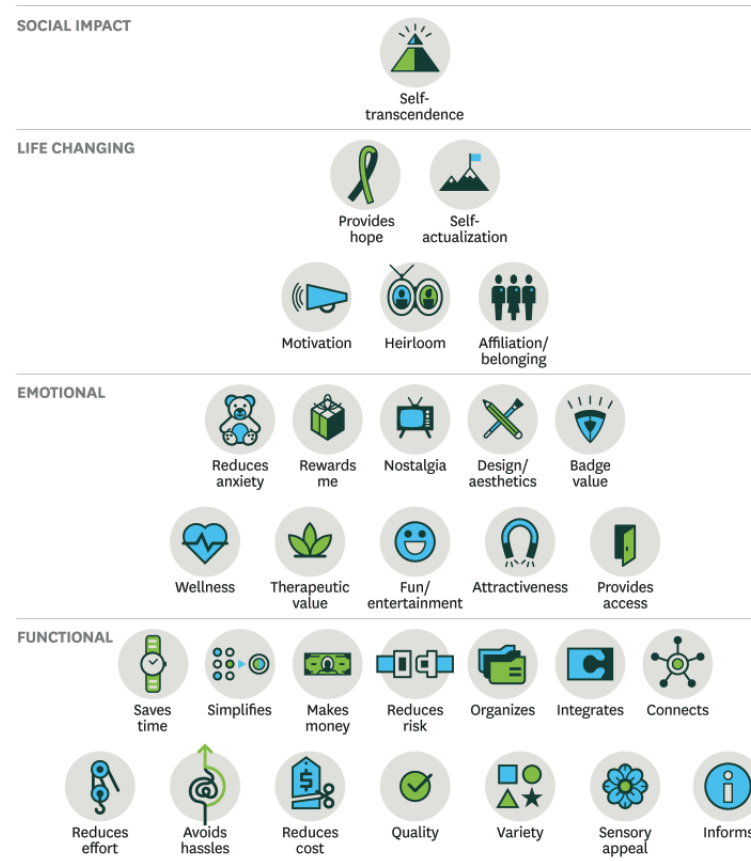


Figure 10: Almquist et al.'s (2016) value pyramid with 30 distinct value elements.

Aside from motivation, the reason the use of a product or a service may come from user requirements. In that regard, the answer may come from the SFS-EN ISO 9241-11 (2018) definition of the context of use components (users, goals and tasks, resources, and environment). Understanding the needs of users and other stakeholders and discovering the requirements set by them, their environment, resources, and other limitations, brings us closer in gaining a more holistic insight of why the object of inspection is being used.

On the other hand, one could inspect the reasons from a negative perspective: why they do not use? Sometimes in the middle of development of a product or service, there can be challenges in finding best ways to support user requirements. Understanding reasons for use often indicates somewhat positive experiences. These, however, may not contribute towards understanding missing elements of the current design. Thus, asking why someone would not use your product or service, may open new perspectives in evaluating the motivation, goals, and the missing needs and values in the product or service.

How they use? Understanding how the users currently use or will use the object of inspection provides insight for designing new software or improving existing ones. In designing successful products or services with good usability, designers must support

the goals and tasks of the user in their environment of use and with the resources they have at hand (SFS-EN ISO 9241-11, 2018). If users behave in an unlikely way with the product or service, it may tell of undiscovered or misunderstood user requirements. Defining the user requirements contributes to user understanding in several ways: First, it may provide direct answers to how a certain user, belonging to some user segment, may use the product. Second, it may directly answer which kinds of users may use the product, in other words, providing answers to what user segments exist for the product or service. Understanding user behaviour in the object of inspection can improve design decisions for system improvement, that in turn, may increase usability, user satisfaction, effectiveness, and efficiency (SFS-EN ISO 9241-11, 2018).

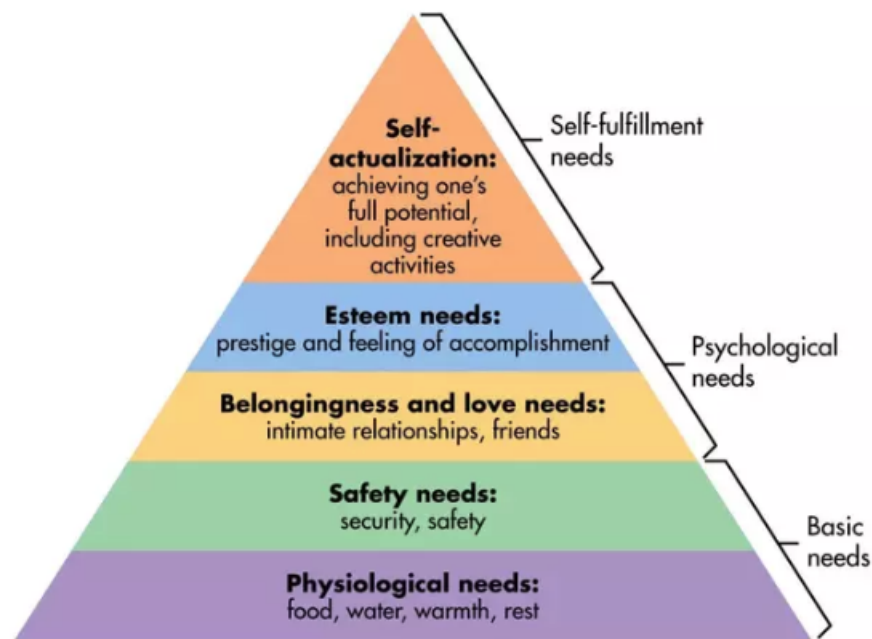


Figure 11: Maslow's "hierarchy of needs" presented in a pyramid model (McLeod, 2018).

3.3 Contribution of the design practices towards user understanding

Different design methods contribute towards gaining a better user understanding. How they contribute depends on when during the development life cycle the methods are being used, how many users can participate in the design activities with the selected design practices, and how successful the design activities are (e.g. how well they were facilitated).

Interviewing with the semi-structured interviews can prove very useful for user understanding discovery and definition. Kvale (1996) refers to two views of interviewing as a research tool: On one hand, they can find specific pieces of information and look for more such pieces, as if mining for high quality minerals. In the empirical study, interviewing is a continuous process of finding small gems of user understanding until the project is satisfied. On the other hand, interviewing can discover new information as interviews go by. So, they are treated as a journey, as jumping from a topic and domain to another provides a chance of learning something significant (Kvale, 1996). Semi-structured interviews allow for flexibility for the interviewers and can elicit complex topics (Wilson, 2013), making the semi-structured interviews very flexible in finding answers to the three main questions of user understanding described above.

Task analysis focuses on understanding users' goals, their tasks and actions, the users themselves, and how their environment affects their tasks and actions (Hackos and Redish, 1998; Courage et al., 2008). Hackos and Redish (1998) suggest to observe users conducting their tasks in the natural environment to discover the true nature of their tasks and goals, which may not come by from an interview with the users; some things are better or easier shown than explained. While observing the users doing their work or interaction with some system, asking questions about users' interactions (as if doing Contextual Inquiry (Thornton, 2019)) can yield more rich feedback than just observing and analyzing. Thus, task analysis can provide great results for understanding the users (Who uses?) and their interactions with a system within a certain environment (How they use?). User requirements and needs (Why they use?) can be captured from interviewing on-site and analyzing their work or interaction.

User stories are often used alongside agile development methods such as Scrum (Deemer et al., 2012). User stories are great tools for conveying pieces of information of requirements, needs, goals, and users involved in a short text (Cohn, 2004; Lucassen et al., 2016), that contributes towards creating user understanding. However, whether user stories can contribute towards providing or eliciting user understanding may depend on the use context. User stories can be formed from existing knowledge and documentation, or based on the decisions of a product owner in Scrum (Deemer et al., 2012). Thus, user stories in this case may provide only existing knowledge from the aforementioned sources or novel ideas from the product owner, either as her own ideas or ideas from other sources conveyed to her. But,

user stories can be crafted together with the real users. For example, inviting real users to individual or group workshops may yield user stories that describe well the requirements, needs, and goals these users in their perceived role can have in the planned or designed system. In this regard, user stories seem to answer well to the questions of who uses and why they use. Due to the nature of user stories not explicitly detailing how a user might go through to reach their goal (Cohn, 2004; Lucassen et al., 2016), they might not provide answer to the last main question, how they use. However, users may explain how they would use a functional requirement, for example, if asked during the crafting of the user story.

Scenarios provide a general description of who are involved in a situation that involves the use of a system, and possible needs, goals, tasks, emotions, and environment related to the use of said system (Bødker, 1999; Carroll, 1999; Rosson and Carroll, 2008). Scenarios tell a story that is usually observed or described by someone, or created by a designer. Scenarios provide good insight, in form of a story, about who are involved in a situation, e.g. additional stakeholders that are involved in the actions of the main user. In addition, these stories can contain information that is not often relevant in other design practices described in this thesis, such as the emotional states of the user throughout the scenario (Carroll, 1999). Scenarios can give detailed information on how the user will approach or use the system, describing the solution to a certain scenario (Cross, 2001). But, they can also be very vague in the solution and describe a situation open-endedly, giving room for future solutions and avoiding 'locking-down' a certain solution path (Bødker, 1999). Scenarios provide generally answers to all the three main questions, and when real users are involved in creating them, there is a possibility for finding more fine-grain description for scenarios.

Prototypes are interactive mediums that give body to an idea or vision of a designed system (Beaudouin-Lafon and Mackay, 2008). As such, they are great for validating existing user understanding or discovering new knowledge in the current user understanding. Prototypes allow for studying the current problem context in the design work (Warfell, 2009; McElroy, 2016) and thus validate whether the existing user understanding that has affected the design of the prototype stands true, when real users are invited to test the prototype. If the users did not approach or use the system as the designer had expected, then the user testing with the prototype could contribute towards discovering new knowledge of the current user understanding and find alternative solutions to the current design (McElroy, 2016). Furthermore, prototypes allow designers to validate or discover the user flows and find answers to how users would use the designed system. But, designers should take into account the fidelity of their prototypes, in how finished state is the prototype and the development of the designed system, to better tailor user tests to find wanted answers in the user understanding. Different fidelity provides different levels of feedback (McElroy, 2016). Tests with prototypes provide good feedback on who uses the system and how they would use the system, but not necessarily provide novel information on why the user would use the system, unless users might disclose reasons for using the system when asked.

Table 19 grades the five design practices, how they provide support to each main question of user understanding.

Table 19: Grading of how each design practice supports user understanding findings relative to the main questions (who, why, and how). *N* = Very low or no support, *M* = Moderate support, *E* = Excellent support. It should be noted that these values are estimations based on the author’s personal experience and literature review. Actual values can be better or worse depending on multiple factors (e.g. quality of facilitation, user representatives, use of internal or external organizational members, organizational politics, research bias, etc).

Design practice	Who uses?	Why they use?	How they use?
Semi-s. Interviews	<i>M</i>	<i>E</i>	<i>E</i>
Task analysis	<i>M</i>	<i>E</i>	<i>E</i>
User story	<i>E</i>	<i>E</i>	<i>N</i>
Scenario	<i>M</i>	<i>E</i>	<i>E</i>
Prototyping	<i>M</i>	<i>E</i>	<i>E</i>

Usage of each design practice can be creative, as long as the focus on conducting and producing outcomes follows the basic principles behind each design practice. Further evaluation of each design practice is outside the scope of this thesis. In defining or increasing user understanding through design practices, the author recommends practitioners to test design practices prior to fully committing into a user understanding study cycle, described in the next chapter.

3.4 Summary of the literature review

In this chapter, the author describes the five different design practices, how they work, and what possible results they could yield. Next is the chapter for the empirical study. It was conducted as an insider action research, as the author had been part of the project. The empirical study employs four design practices: semi-structured interviewing, user story creation, scenario writing, and prototyping. The author had interviewed the case company designer for design practices to be studied in this thesis, thus the five design practices in literature review. Then, the author had judged the four selected design practices to fit the client project he was working on at the time of this thesis. Therefore, those aforementioned design practices are preliminary choices for the empirical study. The following list illustrates the sources used to describe the design practices.

Design practice	Literature source
Semi-structured interviewing	Adams (2010) Beyer and Holtzblatt (1998) Bryman (2016) Hackos and Redish (1998) Kujala and Kauppinen (2004) Kvale (1996) Kvale (2007) Rabionet (2011) Wilson (2013)
<i>A design practice that is flexible and can elicit many aspects from users with good use of probing questions.</i>	
Task Analysis	Annett (2003) Courage et al. (2008) Crystal and Ellington (2004) Hackos and Redish (1998) Hornsby (2010) Holzinger (2005) Norman (1988) Norman (2013) Ritter et al. (2014) Vredenburg et al. (2002) Stanton (2006) Zowghi and Coulin (2005)
<i>Task analysis focuses on understanding how and why the users work with the observable medium (tool, product, or system) in their work environment.</i>	
User stories	Cohn (2004) Lucassen et al. (2016) Deemer et al. (2012)
<i>A short requirement describing template, that is often used to describe the user needs.</i>	
Scenarios	Alexander and Maiden (2004) Bardram (2000) Bødker (1999) Carroll (1999) Carroll (2002) Carroll and Rosson (1990) Cross (2001) Interaction-design.org (2019) Rosson and Carroll (2008)
<i>Scenarios provide real world descriptions of user involvement with a product or a system, in attempt to captivate all affecting aspects holistically (users, their environment, motivation, feelings, etc).</i>	
Prototyping	Beaudouin-Lafon and Mackay (2008) Buchenau and Suri (2000) Camburn et al. (2017) SFS-EN ISO 9241-210 (2019) McElroy (2016) Warfell (2009)
<i>Prototypes allow testing and evaluating of design with relevant stakeholders. Different level of fidelity offers a range of options for testing the design.</i>	

Figure 12, on page 55, presents a model for the empirical study process, that is based on the research process described in Chapter 2. The following Tables 20 and 21 explain each sub-step in the process in more detail.

Table 20: More detailed descriptions of each sub-step presented in the user understanding study process in Figure 12.

Step	Description
1.a.	Define the motivation for conducting the user understanding study (e.g. enhancing sales by understanding how to deliver better value to the end users, gain better feedback from current users, etc).
1.b.	Define the desired outcomes from the user understanding study in relation to the three main questions (who, why, how): to which is the client looking answers for and in what depth? Define the study scope.
1.c.	Manage the client's expectations. Explicitly explain possible outcomes and results, their scope and depth, and the time needed to conduct the study. Be clear, when the results would be ready for presentation.
1.d.	Create sufficient documentation from 1.a and 1.b. Note down the possible length of the study based on planned scope.
2.a.	Understand how the client's product or system works on the part that needs increase or validation of user understanding. Take into account possible external and internal dependencies that may affect some parts of the product or system.
2.b.	Define with the client their current user understanding. Can they answer the three main questions regarding their product or system, and in which depth and breadth?
2.c.	Specify with the client the user requirements for their product or system. Information about the user requirements may provide insight into the context of use.
2.d.	Create sufficient documentation for evaluation and reflection.

Table 21: Continuation of Table 12.

Step	Description
3.a.	Re-visit the areas of user understanding (who, why, how), agree with the client the areas they are interested in, and prioritize the interesting areas of user understanding with the client if necessary.
3.b.	Suggests to the client, which design practices are used to aid the pain areas in current user understanding. Use Table 19 as reference for the perceived effectiveness of each method.
3.c.	Explain the outcomes of the selected design practices to manage client's expectations (e.g. interviews provide a report of findings, task analysis can produce a report of user actions, user stories produce a backlog of user stories, scenarios produce written stories, prototypes can produce an interactable medium).
3.d.	Create sufficient documentation for evaluation and reflection.
4.a.	Agree on a time and date to conduct the selected design practices with the client, client's organization stakeholders, and stakeholders outside the client's organization.
4.b.	Prepare necessary equipment and material for conducting the planned design practices. Also, prepare equipment for documenting the verbal communication and collect any resulting material.
4.c.	Create sufficient documentation for evaluation and reflection.
5.a.	Conduct the selected design practices according to their principles.
5.b.	Document and collect results from conducted design practices with the selected participants for later evaluation and reflection.
6.a.	Evaluate the findings between the results from the client, client's organization stakeholders, and stakeholders outside the client organization, if applicable. If the study was conducted internally, evaluate findings between study participants.
6.b.	Go through the results with the client and the product or system development team (client's organization stakeholders). Evaluate the results and determine the acceptance of reported depth and breadth in analysis.
6.c.	Reflect on the user understanding study, provide self- and organizational learning.
6.d.	Summarize the user understanding study, provide suggestions for the next iteration.
7.a.	Increase of current user understanding or validation of current user understanding.

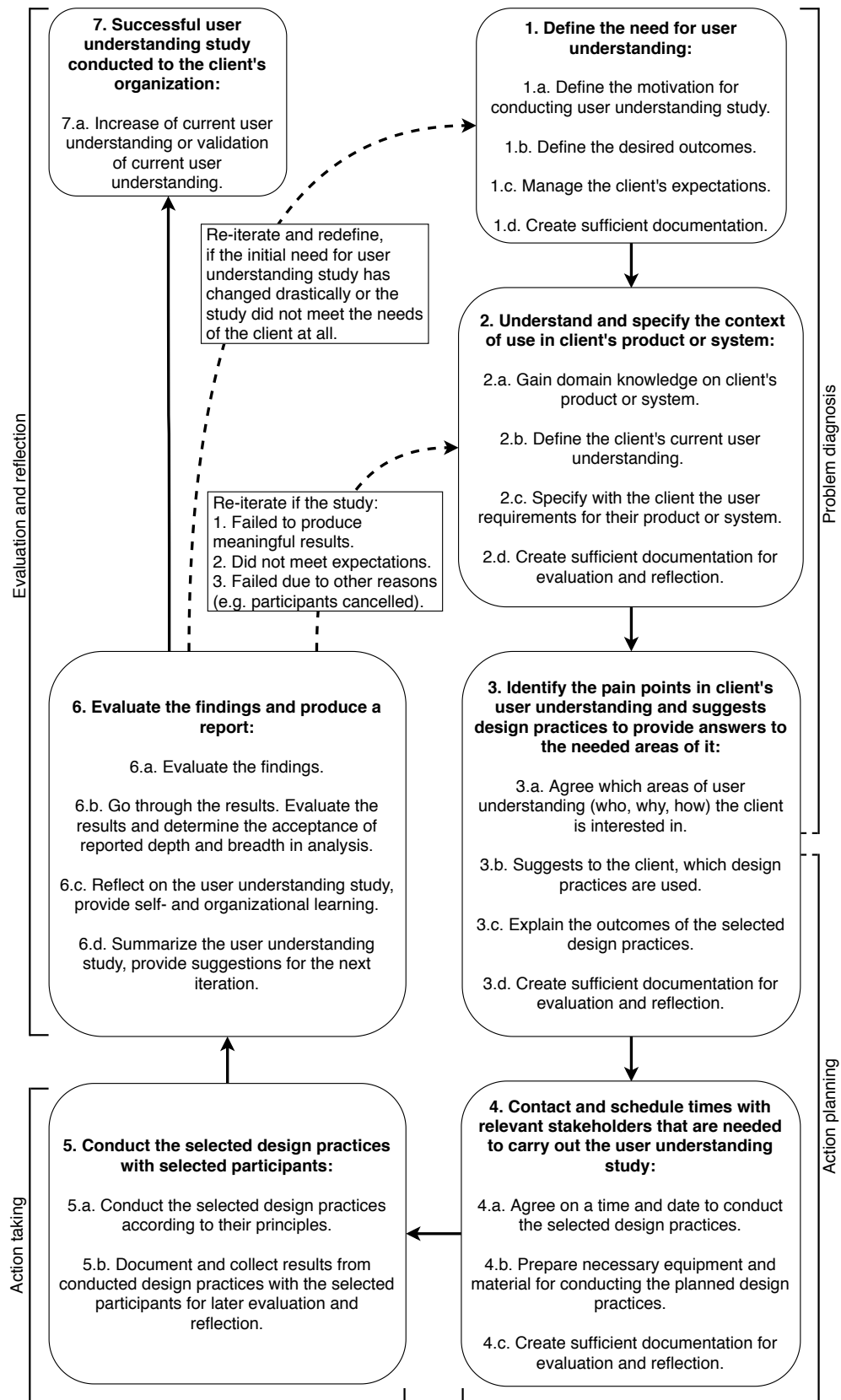


Figure 12: The process of planning and conducting action research to support studying of user understanding.

4 Results of the empirical study

4.1 Problem diagnosis

At the case company, agile software development is practiced by following the agile manifesto of software development:

- individuals and interactions over processes and tool,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation,
- and responding to change over following a plan (Beck et al., 2001).

The product was developed in iterative cycles, where one was 2 to 4 weeks long on average. The development followed basic principles of Scrum. At the time of the product's development, the author was the scrum master of the development team. Scrum was considered for this project, because the development team was 8 by its size. The agile development focused in the following ideas: continuous delivery, working software, quality of work, and customer-centered collaboration.

On the 31st of October 2018, the case company began with the client company the development of their digital coaching platform for marketing employees, called Daily Coach. After 8 months of internal development, the product achieved its Minimum Viable Product (MVP) state. In October 2019, the client company was interested in further developing the product with more focus on finding new solutions for the product through design.

The problem diagnosis was conducted with the product owner from the client company. Motivation for the study was identified as a need for improving the admin panel of the product. In the product owner's opinion, the current admin panel did not offer enough features for it to be a readily sold product. The rest of the product was fine according to the product owner, thus improving the admin panel was seen as the highest priority, the solution to the challenge of marketing the product, and as a goal for this study. As for the definition of "improving the admin panel", what it meant, the author proposed eliciting user requirements as the outcome of the study, as they would speak out what must be done for improving the admin panel.

The product owner's current user understanding consisted of knowing their current and planned user groups for the product. Little knowledge was for defining the reasons or motivation for the use, and behaviour in the use of the product. For this study, the product owner wanted insight into the currently lacking parts of the user understanding.

The study started 13.11.2019. The desired schedule for this study was set to a maximum of one month, give and take a week from the starting date. The desired scope was to use minimal effort possible while producing meaningful results. The case company has done in the past design work that touched a certain part of a

system and lasted up to a month, sometimes faster. The study conducted to the product owner was an activity that could be comparable to the design activities offered to other clients of the case company, in a sense that the work was preparatory for iterative development efforts.

A work day was used to map the current state of the product. After that, the author had mapped with the product owner their topics of interest within the product for finding relevant user requirements or for categorizing them. The topics of interest within the product were listed as:

- Communication to users
- Answers from users
- Comments from users
- Content management
- Course management
- Prizes
- Admin functions
- Points needed for prizes

Figure 13 presents the home page of the admin panel, Figure 14 presents the first page of creating a course in the admin panel, and Figure 15 presents the second page of creating a course. Total effort used for problem diagnosis is presented below in Table 22.

Table 22: Used effort in problem diagnosis phase.

Activity	Used time in hours
Familiarize self with the project in more depth	6 hours 30 minutes
Create a presentation set for problem diagnosis meeting	3 hours
Agree on a meeting time with the product owner	15 minutes
Introduction to action research and user understanding	1 hour
Problem diagnosis meeting with the product owner	30 minutes
Total	11 hours 15 minutes
Product owner's effort	1 hour 45 minutes

Due to the simplicity of the product and its features, it was relatively fast to go through the project. Creating the presentation set that would summarize on an acceptable level the concept of action research and user understanding took almost half the amount effort compared to familiarization of the project.

Status	Course name	Company name	Company email	Participants	Questions	
<input type="checkbox"/>	running	Refined Refined Plastic Ball	Collier and Sons	noreply@testdailycoach.com	12	27
<input type="checkbox"/>	running	Tasty Awesome Granite Sausages	Bode, Muller and Schuppe	noreply@testdailycoach.com	29	31
<input type="checkbox"/>	running	Fantastic Small Plastic Towels	Flatley - McClure	noreply@testdailycoach.com	19	24
<input type="checkbox"/>	running	Intelligent Generic Concrete Gloves	Larkin - Kuvalis	noreply@testdailycoach.com	17	42
<input type="checkbox"/>	running	Refined Licensed Concrete Soap	Dooley, Windler and Fadel	noreply@testdailycoach.com	22	26

Figure 13: The state of the admin panel during the thesis. The view is from the homepage of the admin panel.

Course

Prize

User

Question

Answer

Comment

Import

←

GENERAL

THEME

OPTIONS

Course name *

Company name *

Company email *

Sender name *

Details *

Course end date *

Footer *

Status *

SAVE

Figure 14: The first page of creating a new course in the product. Many fields contained information that was not clear even to the internal users of the admin panel, making it challenging to introduce the course creation process to new internal employees.

The screenshot shows a web interface for creating a course. At the top, there is a green header bar with a hamburger menu icon, the text 'Create Course', and 'DAILY COACH' on the right. On the left, a sidebar lists various course components: Course (selected), Prize, User, Question, Answer, Comment, and Import. The main content area is titled 'THEME' and contains five input fields for color settings: 'Logo URL *', 'Background color *', 'Accent color *', 'Main text color *', and 'Accent text color *'. A green 'SAVE' button is located at the bottom right of the form.

Figure 15: The second page of creating a new course in the product. Some colour related settings were confusing or not readily understood to internal users.

4.2 Action planning

In action planning, the author proposed four different design practices to be used in the study. These design practices were chosen based on the product owner's need in deepening their user understanding: the motivation for the use of the product (why they use?) and the behavioural data of product usage (how they use?). For action planning, it was important to agree on the amount of users and the effort used to stay in the desired schedule and scope.

At least 5 participants were planned for the study, as Hackos and Redish (1998) suggest at least that number to produce sufficient results in qualitative research. Then, the author planned to carry two sessions per participant. Each session would be 1-to-1 with the participant. Semi-structured interviews were planned as preparatory sessions for later workshops. Then, the workshops were a second session that would vary between user representatives and the rest: user representatives would craft user stories and prototype, and the rest would do scenarios and prototype. With such design practices planned, the product owner would get results from the interviews, a list of user stories, and a collection of scenarios. Prototyping in workshops was planned as a supporting activity for both user story and scenario crafting.

Interviews were planned as semi-structured interviews. The interview template was constructed according to Wilson's (2013) template. The same interview template was used to plan all interviews. Questions of the interview were designed to probe answers from interviewees of their perceived important and useful features of the product, either what already exists or what would be nice to have.

Table 23: A semi-structured interview guide by Wilson (2013).

Activity	Comments/Questions	Approximate Time
Introduction	Brief the participant. Introduce self. Explain goals of interview. Review interview method, use of data, confidentiality, so on.	10 min
Structured topics	Topic 1: Background Question 1a Probe 1 Probe 2 Probe 3 Topic 2: Context of Work Question 2a Probe 1 Probe 2 Question 2b Probe 1 Topic 3: Use of Product Question 31 Probe 1 Topic N: Additional topics	40 min
General questions and open dialogue with participant		30 min
Closing comments and completion of any paperwork (receipts, final questionnaire, etc.)		10 min

User story crafting and scenario crafting workshops were planned similarly. In both workshops, prototyping was used to help participants grasp the true nature of the product's current state and to see for themselves, how different features and actions would work. The workshops would support general understanding of the product and finding important and useful features for its current and future state. The workshops were planned to follow a simple format:

1. There would be an introduction and general knowledge about the design practice to be used (user stories or scenarios)
2. Then, the author would have a short practice in creating either user stories or scenarios with a commonly known theme (e.g. user stories about a bus trip)

from point A to B, or a scenario of leaving to work from home)

3. The interesting topics defined in problem diagnosis would be shown to the participant and they are asked to craft the first set of user stories/scenarios
4. After they have done some, the participant gets to prototype the current product and try the Admin features in the admin panel themselves.
5. Then, after prototyping, the participants gets to crate more user stories/scenarios
6. And finally, the session is wrapped up and closed with a short feedback conversation

In the workshop plan of using user stories, those followed Cohn's (2004) template of user stories modified to: "As a (role), I want (function) so that (business value)". For scenarios, the format was planned to be an interactive conversation, where the author allowed room for the participant to craft their scenarios, but asked probing or defining questions, if the scenarios seemed to be too open or not detailed enough. Prototyping was planned to be straight forward: the participant would get a general overview of all relevant admin panel features and actions from the author in a test environment, and then the participant would be given a chance to try the admin panel themselves as much as they liked within a set time frame.

These sessions were planned to last for an hour each, meaning a total of 2 hours per participant. The author scheduled the session times with each participant in a way that there would be at least a day in between the first and the second session, to avoid cognitive exhaustion.

Each activity to be conducted in action taking were planned to be as easy as possible for all participants, so that partaking would be easy. Each activity was thus planned in such a way that there would be no pre-study material required or any preparations: the participant could just walk in as they were and partake in the study with minimal effort (e.g. taking just their laptop or mobile phone with them).

These plans were presented to and accepted by the product owner. The total effort used for action planning so far was 2 hours and 45 minutes. Overall estimated effort for conducting the plan was 12 hours 30 minutes.

Table 24: Used effort in action planning phase.

Activity	Used time in hours
Evaluate the problem diagnosis results	1 hour
Create a presentation set for action planning meeting	1 hour
Agree on a meeting time with the product owner	15 minutes
Action planning meeting with the product owner	30 minutes
Total	2 hours 45 minutes
Product owner's effort	45 minutes

Defining the needed user understanding with selected design practices $(1.00 \text{ h} + 1.00 \text{ h}) \times N$	User 1 + Henri (1.00 h + 1.00 h) User 2 + Henri (1.00 h + 1.00 h) User 3 + Henri (1.00 h + 1.00 h) Product owner + Henri (1.00 h + 1.00 h) Project manager + Henri (1.00 h + 1.00 h)
Analysis and reporting of findings from each session $(0.25 \text{ h} + 0.25 \text{ h}) \times N$	About 15 minutes of wrapping each session and producing short summaries
Evaluating results after action intervention 1.00 h	Henri goes through with the product owner
Total $2.00 \text{ h} \times 5 = 10.00 \text{ h}$ for planned sessions $0.50 \text{ h} \times 5 = 2.50 \text{ h}$ for analysis and reporting Grand total: 12.50 hours	

Figure 16: Estimated effort for conducting the action plan. Caption from author's PowerPoint-presentation.

4.3 Action taking

After action planning, the action taking included conducting the selected design practices with the planned users and collecting the results. Action taking proceeded according to the plan in action planning.

The author recruited 5 participants to the study. All participants were either from the case company or the client company. Of the 5 participants, 3 of them were presenting regular users, and the others were the product owner, and the project manager. The 3 users had not used the product before but may have heard of it from their colleagues. The product owner and the project manager had planned features and tested some. They were treated as somewhat more knowledgeable users.

4.3.1 Semi-structured interviews

The first session with each participant began with a semi-structured interview. Interviews were held in a conference room with the participant, making sure they had enough privacy and all necessities for starting the interview. The author used his OnePlus 6 phone to record speech, a personal notebook for taking notes, and some printed pictures of the admin panel during the interviews. The pictures of the admin panel were used to ask the interviewee, how they would describe the items in the picture and how they perceived some functionality to be done by just looking at a static image. Figures 17 and 18 present the template crafted and used in the interviews, and Figure 19 shows a picture used in the interview. A consent form

was prepared for the study participants. However, the form was never used, as all participants deemed it unnecessary and trusted the author, as the participants were all from the same office space, despite coming from different companies (from the case company and the client company). The consent form is based on the author's other consent forms used in qualitative research, which he had used in his studies.

Each interview lasted an hour at most. To ensure this, the author used the recording timer to measure interview time. If at any moment the timer showed 45 minutes had past, the author would then move on to the final questions and end the interview with feedback discussion. In the feedback discussion, the recorder was left on to capture additional information not collected through interview questions (as suggested by Bryman (2016)).

Table 25: Used effort for constructing and conducting the interviews.

Activity	Used time in hours
Preparing the interviews	2 hours
Interviewing user 1	55 minutes
Interviewing user 2	45 minutes
Interviewing user 3	50 minutes
Interviewing the product owner	55 minutes
Interviewing the project manager	40 minutes
Total	6 hours 05 minutes

Semi-structured interviewing Template by Wilson (2013), modified for Henri's Master's Thesis

Activity	Comments/Questions	Approximate Time
Introduction	<p>Brief the participant. Introduce self. Explain goals of interview.</p> <ul style="list-style-type: none"> Finding answers to 'why' and 'how' the interviewee would think the product Daily Coach product is used by target audience <p>Review interview method, use of data, confidentiality, so on.</p> <ul style="list-style-type: none"> Introduce the consent form if necessary Introduce the right to withdrawal from study Anonymity of answers 	10 min
Structured topics	<p>(I) Introduction of the product: Daily Coach</p> <ul style="list-style-type: none"> Can you provide a short summary of Daily Coach as a product? <ul style="list-style-type: none"> Main highlights or points of the product? Can you list what the Daily Coach is for? <ul style="list-style-type: none"> What is it? What does it do? What does it offer? What thoughts should the product name or the concept invoke in the target audience? <ul style="list-style-type: none"> Functionalities? Marketing? Values? <p>(II) The product: Daily Coach</p> <ul style="list-style-type: none"> What do you think is important for Daily Coach? <ul style="list-style-type: none"> What would be important features for Daily Coach? In what situations could you see Daily Coach being used? What value would you expect Daily Coach to deliver? (Show the Value Pyramid if they need help) 	25 min

Figure 17: The interview template for semi-structured interviews, page 1 of 2.

	<ul style="list-style-type: none"> • What do you expect Daily Coach to do? <ul style="list-style-type: none"> ◦ Goals ◦ Functionality ◦ Expectations ◦ Service <p>(III) Functionality of the product</p> <ul style="list-style-type: none"> • An example: Think of a car. It offers transportation and has multiple seats. It has an engine and four wheels. It can be manual or automatic. It can go forward or reverse. • I had just described some of the functionality and characteristics of a car. Within similar manner, could you describe your opinion of functionalities and characteristics of the Daily Coach? 	
Product images and reactions	<p>Show image N: How would you interpret the image based on the following questions:</p> <ul style="list-style-type: none"> • Describe the view in front of you. <ul style="list-style-type: none"> ◦ What functionalities you believe you can identify? ◦ How would you conduct _____ in this image? 	10 min
General questions and open dialogue with participant	<p>How did you feel about taking part in this interview?</p> <p>Do you feel that you could take part in these interviews in the future?</p> <p>If there would be iteratively similar sessions in the future within certain intervals of time, would you take part in these?</p> <p>What feelings this interview left to you?</p>	10 min
Closing comments and completion of any paperwork (receipts, final questionnaire, etc.)	<p>Demographics:</p> <ul style="list-style-type: none"> • Age group within 10-year accuracy • Have you used anything similar in the past or currently? 	05 min

Figure 18: The interview template for semi-structured interviews, page 2 of 2.

The interview questions focused on finding important and meaningful user requirements for the current and future version of the product. The questions were also constructed to support finding user requirements that supported finding answers to the relevant question of user understanding: the *why they use* and *how they would use* questions. Below are some interview answers from the participants:

- Q: Can you provide a short summary of Daily Coach?
- A: It's a digital tool for change management and learning after a project
- Q: What do you think is important for Daily Coach?
- A: Easy to use, provides daily reminders, creates community feeling, and collect opinions
- Q: What do you expect Daily Coach to do?
- A: It provides immediate, real time feedback. Supports and coaches organizational change.

Status	Course name	Company name	Company email	Participants	Questions
running	Refined Refined Plastic Ball	Collier and Sons	noreply@testdailycoach.com	12	27
running	Tasty Awesome Granite Sausages	Bode, Muller and Schuppe	noreply@testdailycoach.com	29	31
running	Fantastic Small Plastic Towels	Flatley - McClure	noreply@testdailycoach.com	19	24
running	Intelligent Generic Concrete Gloves	Larkin - Kuvalis	noreply@testdailycoach.com	17	42
running	Refined Licensed Concrete Soap	Dooley, Windler and Fadel	noreply@testdailycoach.com	22	26

Figure 19: The home page of the admin panel was used in the interview to ask the interviewee to describe visible functions or features for the user, and how they would execute some action, such as adding a new course to the database, adding participants to a course, or importing a course from a comma-separated values file.

After the interviews, the author held the workshops. They were either a user story crafting or a scenario crafting workshop, where the former was for users and the latter for the project manager and the product owner.

4.3.2 User story crafting workshops

The workshops held for the participants contained user story crafting and prototyping. In the workshop, the participants were taught the concept of user stories, practiced making user stories themselves, tested the current implementation of the product to understand the context of the product, and finally made some more user stories. The flow of the workshop was handled by a PowerPoint presentation, with slides presenting the agenda of the workshop, individual tasks, and links to the relevant Google Drive files.

The workshops began with an introduction, welcoming the participant, recapping the data handling procedure, and asking for consent. In the workshops, there was also no need for consent form, as it was given verbally and there was trust between participants and the author. The author had asked each participant to bring their laptop with them, so that the user stories could be created and stored in a shared Google Sheet file. The author took notes only to his personal notebook.

The participants were not required to have any prior knowledge about user stories. At the beginning of the workshop, the author introduced the basic format of user stories provided by Cohn (2004). Then, the author proceeded to conduct a short practice with the participant in crafting user stories by using public transport as an example. The example was: "As a public transport user in the capital region, how would you describe user stories related to using public transport?". The author created a couple examples to the participant, to let them get the gist of it, before they got to create some themselves.

After the introduction, the participant got to do their user stories. At this point, they had not seen the whole system before, apart from some images used in the semi-structured interviews. After the participants had done their first batch of user stories, the author presented the product and allowed them to test out some features if they wanted. After the author's presentation or testing, the participants proceeded to make more user stories. The below list summarizes the agenda of the workshops and allocated time, Figure 20 shows a random pick of created user stories in each planned category, and Table 26 shows total effort for each workshop.

1. Introduction to the workshop and examples (10 minutes)
2. Crafting user stories (15 minutes)
3. Testing the current state of the admin panel (10 minutes)
4. Crafting more user stories (15 minutes)
5. Summary of user stories and open feedback (10 minutes)

Communication to users	Content management	Course management	Admin functions
As a course member, I'd like the course e-mail to find its way to my inbox instead of spam, so that finding the e-mail would be easy	As a course member, I'd like to be able to edit my answer in case I choose an answer option by mistake	As an admin, I'd like the system to double check my actions that change course content, so that I don't accidentally make critical changes too easily	As a course member, I'd like to add my own course content to the courses I'm attending, so that I could share my knowledge to others
Answers from users	Comments from users	Points needed for prizes	Prizes
As a course member, I'd like to see other's answers only after having answered myself, so that my learning is not affected by others	As a course member, I'd like to be able to have an open conversation in a chat widget with other participants, after I had participated in today's question	As a course participant, I'd like to get extra points for reminding other course participants about answering the today's question, so that Daily Coach could create community feeling between course members	As an admin, I'd like to share the best performing member of the week in social media easily from the admin panel

Figure 20: A caption of user stories made by workshop participants.

Table 26: Used effort for preparing and conducting the workshops.

Activity	Used time in hours
Creating the workshop template	1 hour
User story workshop for user 1	45 minutes
User story workshop for user 2	55 minutes
User story workshop for user 3	50 minutes
Total	3 hours 30 minutes

4.3.3 Scenario crafting workshops

The scenario crafting workshops were similar to user story crafting workshops by the agenda, with the exception of the participants (the product owner and the project manager) crafting scenarios and not user stories. Participants were not required to know about scenarios or crafting scenarios before hand. During the workshop, the author presented an example scenario and posed some questions to enrich the content. The example scenario was:

45-year-old Pekka leaves from his home for work at 07.15. He travels to work with his 1995 Volvo car. His workday begins at 08.00 in the morning and travelling to work takes usually 30 minutes. This morning, Pekka can't leave for work at his usual 07.15 schedule, because it's too cold outside and the car won't start from the apartment house's parking lot.

The example scenario gave details about the actor and something about the background and situation, but little information of what could be done. An open-ended scenario provides possibilities for further refining to find solution concepts (Bødker, 1999). Thus, it was possible to create probing questions and the author presented some probing questions and asked the participant to think of some as well. Example questions were:

- Could Pekka have anticipated the weather somehow to prevent this situation from happening?
- Could Pekka use some other mode of transportation than his own vehicle to get to work on time?
- How could coming late to work affect Pekka? Does he have a set working time?

After practicing, the participant then made some scenarios themselves. After the initial batch of scenarios, the author showed to the participant the product and allowed them to test some functions and features themselves. After that, the participant made some more scenarios, until it was time for feedback discussion. In essence, the format was same as with user story workshops, with the following agenda:

1. Introduction to the workshop and examples (10 minutes)
2. Crafting scenarios (15 minutes)
3. Testing the current state of the admin panel (10 minutes)
4. Crafting more scenarios (15 minutes)
5. Summary of scenarios and open feedback (10 minutes)

As the participants made scenarios, the author would ask probing questions if the scenario seemed too vague and the participants agreed. Below is an example scenario made by one of the participants.

50-year-old sales manager Kaisa wants to know whether during the project the newly developed processes and procedures have been taken to use by the sales team (8 people). She logs in during her morning coffee into Daily Coach admin panel and checks the last Friday's (it's currently Monday) survey answers from the sales team, that concerns the newly developed processes and procedures. The admin panel shows that 7 out of 8 have given answers to the survey. So, Kaisa decides to contact the team lead of the sales team and asks them to remind that one person, who hasn't answered the survey, so that the person would become familiar with the newly developed processes and procedures by taking the survey.

Despite the times agenda, both workshops went overtime. When the author spotted the risk for overtime, he kindly gave an option to end workshops on time despite them being in progress, or allow it to go overtime if the participants agreed to.

Table 27: Used effort for constructing and conducting the scenario crafting workshops.

Activity	Used time in hours
Preparing the workshops	2 hours
Scenario workshop for the Product Manager	1 hour 10 minutes
Scenario workshop for the Project Manager	1 hour 5 minutes
Total	4 hours 15 minutes

4.3.4 Prototyping

Prototyping was used in semi-structured interviews and both workshops. In semi-structured interviews, prototyping of the admin panel was done by printed screen shots of the system and by asking the participant to describe the functionalities and features they saw in the static image and some feature would happen in their opinion. The author suggested, but did not limit, the participants to just explain verbally and possibly point with their finger some elements in the static image. Had the participant have some trouble recognizing some items due to printing quality, the author would then explain the content.

In both workshops, prototyping the admin panel was done by showing the test environment of the admin panel from the author's work laptop. The author presented all features of the admin panel to the participant and explained them. As there weren't many features, showing all features was possible. For a larger scale product, it should be taken into consideration how many features can be prototyped to keep workshop sessions short enough. After the author's presentation, the participant was given the chance to test functionalities themselves and ask questions from the author at the same time.

4.3.5 Total effort of action taking

In total, the author conducted 5 semi-structured interviews, 3 user story crafting workshops and 2 scenario crafting workshops. From the user interviews, each question yielded at least one answer, and a total 63 answers, excluding the answers to images. Images were used to report challenges with the current admin panel and some challenges, such as creating a new course, were reported. Scenario workshops yielded both 6 scenarios in total. The scenarios were converted into user stories, and combined with user story crafting workshop results, all workshops resulted 67 user stories. It should be noted, that the numbers do not exclude duplicates or similar results.

The author spent in total 12 hours and 45 minutes in the action taking phase. The following table breaks the total used effort for participants in the action taking phase.

Table 28: Used effort for each participant in action taking phase.

Participants	Total effort for action taking
User 1	1 hour 40 minutes
User 2	1 hour 40 minutes
User 3	1 hour 40 minutes
Product owner	2 hours 5 minutes
Project manager	1 hour 45 minutes
The author	12 hours 45 minutes

4.4 Evaluation of the design practices

In evaluating the research results, the author had two goals: First, to summarize the results into an easily understood form, which he would then refine into information that could provide self and organization learning. Second, for the insider action analysis cycle to be complete from the client's perspective, the author evaluated the results with the product owner to seek passes or fails against evaluation criteria.

4.4.1 Reflective learning

Reflective learning was formed by collecting feedback during action taking by asking a set of questions after the interviews and workshops. The questions were:

1. How was this session?

2. If you had to describe this session as a scale between easy and hard to attend, how would you describe this session?
3. How was the length of this session?
4. Would you participate in these sessions in the future, e.g. to help product development?
5. How were the used design practices? Did you find them useful or interesting, or something else?
6. Did you gain something from this session?
7. Anything else you would like to say?

How was this session? In the interviews, all participants expressed positive emotions and answers. Main highlights from feedback was the feeling of participating and contributing to good cause, the feeling of their opinions having possible positive impact, learning something new from the session, and having the chance at being interviewed (especially for those who had been often in the interviewer position). Some expressed the feeling of accomplishment and exceeding of their own expectations. However, only negative comment about the interview sessions regarded the questions and the theme: the goal or the setting was not clearly explained and some questions seemed leading or lacked breadth. It is possible that the author had failed a routinely introduction part of the interview and skipped some parts, as he made assumptions of the interviewee based on how he knew the person.

In the workshops, crafting user stories were seen as a pleasant and fun organized activity. This feedback was general amongst those who had not done such activity before, thus being possibly a general positive reaction to something new and fancy. Scenario crafting workshops were seen as useful in creating the product vision and testing the ability of a person to describe their idea of the product. For example, scenario crafting could be used to benchmark past ideas with current ones and comparing differences to business vision, how it has developed through time. Another note on scenario crafting workshops was how easily just a few sentences could initiate such deep conversation and reflective thinking. The author made prompting questions to the participant, when they seemed to get stuck or left some parts in the stories too vague.

If you had to describe this session as a scale between easy and hard to attend, how would you describe this session? In the interviews and workshops alike, these sessions were deemed mostly easy to attend. The author had not required any pre-study or any preparations from the participant, with the exception of workshops where the author asked participants to take their laptop with them. Some challenges or worries were expressed. For user story crafting workshops, it was said that having a single category instead of many could have made it more easier to come up with user stories. Truly, making the scope smaller may produce more results in a single category compared to how many would be created with a larger scope, and for individuals who may not be strongly affiliated with the product, holistic

view of the product may prove difficult. Then, in scenario crafting workshops, one concern was the concept of scenarios: some claimed that scenario crafting may not be a familiar concept for individuals who are not strongly affiliated with the product (e.g. non-steering group stakeholders). They might find some challenges in creativity, creating descriptions that could match the real world as much as possible (not going overboard with fiction), and finding the optimal depth for descriptions for meaningful descriptions.

How was the length of this session? In both interviews and workshops, the lengths of the session was deemed fitting. For all sessions, the author planned for an hour sessions so that an interview and a workshop wouldn't be on the same day for the same participant. This decision seemed correct, as after an interview or a workshop the participants often seemed express facial emotions or speech that would suggest such mental exhaustion that continuing with another sessions could have been unfruitful. Despite going a little overtime (5 to 10 minutes) with scenario crafting workshops, this did not affect the participants opinion on the length of the session.

Would you participate in these sessions in the future, e.g. to help product development? None of the participants expressed any negative comments and were happy to continue with participating to these events in the future, had this been an actively used iterative and continuing process. It is difficult to say with just one cycle of this empirical study, whether the answer was biased and political, which could be studied more with more iterations and comparing participation activity. Or, the answer could be a polite and political answer to the author as all participants worked under the same roof, or it could have been an honest answer from all participants.

How were the used design practices? Did you find them useful or interesting, or something else? The semi-structured interview sessions were quite known to the participants; they were not strict conversations and seemed to not be completely made up on the spot. All participants had been in semi-structured interviews before. Reviewing of static images of the admin panel user interface had been a pleasant and refreshing experience to the participants, as they found it interesting to think about different possibilities that could happen after doing something in the image. User story crafting scenarios, on the other hand, sparked interest in some. They had been interested in trying out crafting user stories during user interviews, which they do regularly in their work. For scenarios, some participants thought of using them in their consulting work with their customers, to aid product visioning and benchmarking it throughout a project.

Did you gain something from this session? Answers to this question were quite similar as to the above question. In interview session, design practice wise there was nothing new, but the participants walked away from the session with a more positive feeling than what they had when walking in, although more mentally exhausted. Similarly in both workshops, the participants were more mentally exhausted in the end, but felt positive through their accomplishments with user stories or scenarios.

Interests towards both design practices seemed to carry outside of the study to their personal work, based on their answers to the previous question.

Anything else you would like to say? With one participant, a workshop session was held at the end of their work day, close to 4 PM. They wished for the workshop to be held in the early morning, instead of at the late afternoon. Taking into consideration that an hour workshop may be intensive and cognitively exhausting, the author may have not made the best decision to have held a workshop in the late afternoon.

4.4.2 Client's evaluation and acceptance of result

With the session feedback put into a readable form, the author went through them with the product owner. The author had gone through the same points pointed in above section in a more compact version. Then, the outcomes of the insider action research were looked at: a total of 67 user stories from crafted user stories and refined from scenarios 6 scenarios. Similar or almost identical user stories were grouped together, so that it could be more easily seen which stories were unique or distinct and which were highlighted by several participants. A user story which had other similar ones could tell of its importance or how common it was to the users.

To recap, the product owner, in the role of the client as well, had expected user requirements from the study that could improve the admin panel of the product. Their current user understanding was strong in the field of understanding who their users were or could be. They were not so sure about why their users use or would use their product, or how they use or would use the product. The author selected such design practices that he believed to help in providing answers to the wanted areas of user understanding, and which would satisfy the product owner's expectations.

The study resulted user stories and scenarios, which describe well the user requirements. The author went through the user stories and scenarios with the product owner, to check that the content in the user stories and scenarios were meaningful and reasonable. The user stories and scenarios both described the current requirements and future requirements for the admin panel, thus fulfilling the need for admin panel improving user requirements. And finally, the insider action research study was concluded in 20.12.2019.

The acceptance evaluation criteria had three items: First, did the study produce meaningful results to the client? Second, did the study produce such results which met with the expectations? And third, was there any reason to call this study a failure and what was the reason? The study did not exceed the planned schedule and the product owner was content with the outcomes. As there were no complaints or any other reason to deem the study failed, the study was concluded successful.

In total, the evaluation took 4 hours of analyzing the outcomes and preparing the presentation material, then evaluating them with the product owner took an hour.

Table 29: The author's and product owner's used effort in total for the whole study.

Activity	Used time in hours
Problem diagnosis effort	
Author's effort	11 hours 15 minutes
Product owner's effort	1 hour 45 minutes
Action planning effort	
Author's effort	2 hours 45 minutes
Product owner's effort	45 minutes
Action taking effort	
Author's effort	12 hours 45 minutes
Product owner's effort	2 hours 5 minutes
Evaluating research results	
Author's effort	4 hours
Product owner's effort	1 hour
Total	
Author's effort	30 hours 45 minutes
Product owner's effort	5 hours 35 minutes

For future cases, if insider action research would have been a continuous activity alongside agile software and product development, the next step would have been defining the motivation and needs of the study and selecting relevant design practices to support the motivation and expected outcomes.

4.5 Lessons learned from the design practices

The case company and the client company had not been involved in insider action research studies before this. Thus, this study was able to contribute lots of organizational learning to both companies. This specific case provided information on how insider action research could benefit or worsen the product and software development of a project that has reached an MVP state.

Semi-structured interviews helped in defining the why and how of user understanding. The interviews with the participants focused in discovering and exploring potential motivations and values for the current and future product use. Probing questions were designed to explore behavioural aspects or direct answers to

how some valued features would be used. The strategy for interview questions was to discover motivating or valued features (understanding why something is needed, why it would be used), and then ask defining questions of how such feature would be used.

When planning the interview questions and probe questions, one could ask themselves whether their questions answer to any of the three main questions of user understanding: who uses, why they use, and how they use? Interviews can produce great results, but require the avoidance of the following pitfalls: First, for example, a badly planned interview where the questions do not provide answers for the wanted area of user understanding. Second, if the interviewee lacks experience in participating in interviews or has lacks social skills for interviews (e.g. a reserved personality can help in avoiding giving body signals to the interviewee, but may be lacking in using probing questions unless they are sure the question can provide results). And third, sometimes preparatory work is best as it could be, but the recruited interviewee might be representing the wrong audience or they might be shy in giving answers. When recruiting participants, this risk can be mitigated by surveying recruited people before inviting them to the interview, to make sure that they can contribute to the interview and feel that it was worth their time.

Interviewing the participants about systems improvement can create a foundation for other design practices to work on. Semi-structured interviews provided an opportunity to start the improvement work for the product. Various questions eliciting the current strengths and targets of improvements for the product's future state had prepared the interviewees for upcoming design workshops. Despite the interviewees not necessarily remembering exactly the topics and ideas they discussed in the interviews, at least the design workshops did not appear to come as a surprise to any participant. Although, many had refined some of the ideas they had explained in the interviews into user stories or scenarios, depending which design workshop they had attended.

User stories helped in discovering the users and their motivations for their needs. User stories were used in a workshop where the focus was in finding user requirements for the current and future state of the product. Finding the user requirements meant understanding both the current state and its limitations, as well as the future state and what could be possible for the product. Many of the user stories began with specifying the user, who it was. Thus, in creating the user story, the first focus was in specifying the actor, who, in user understanding. Then, the need of the user, which often included the business value that sometimes described the motivation or value of the need for the user, which is the essential part of answering the "why" in user understanding.

User stories are great in defining the who and why area of user understanding. In crafting user stories, one must think of who is acting in the story and what is their need or motivation. The resulting user story then describes the actor, who represents some relevant user group, and the need, what is wanted by that user. If the need is well understood, then it can be fairly easy to describe the motivation

behind the need. Motivation can be e.g. a necessity (work cannot be completed without using this product), gained value (using the product makes my work less stressful), avoidance of lost value (not using this product might leave me out of community), or emotional impact (using this product makes me feel a better person). However, user stories and their quality of contents depends on the written skills of the creator. How well can the person depicting the user story create such content that it contributes towards increasing the user understanding? Also, the background of the person may affect how well they can create realistic or fitting user stories according to the user understanding needs. A real user might be able to describe their needs and motivations, but if they represent a user group that is not relevant, their user stories may end up useless for the needed user understanding.

User story crafting can be easy and produce a good amount of user stories within an hour. Crafting the user stories with the participants was perceived as an easy activity. Although some may have had some trouble in creating user stories in all asked categories (8 distinct categories were presented), participants were able to generate 16 user stories on average. However, the number of user stories does not tell the truth of the quality of user stories. Some user stories depicted the existing user requirements, thus the actual amount of user stories that described improvements and problems of the product for its system improvement were less than the reported average. The user story crafting efforts may have benefited from a set view, e.g. focusing only in improvement related user stories, that change something existing or demand implementation of a new feature or function. But, the author did not set a such strict requirement, so that the participants could get more freedom in creating user stories. In general, depending on whether to focus on allowing the creator of user stories to experience with user stories, and that way gain experience in creating them. Or, set a clear view of what kind of user stories are wanted, such as strictly system improvement user stories, bug fixing user stories, or some business value oriented user stories.

However, user story workshops may be difficult to conduct, if there are conditions, such as a set goal for minimum amount of user stories produced per workshop, that may affect whether a workshop can finish on time. It is important to respect the time limitations of the participant, but if the workshop is cut off due to time limit and little effort was made, both the participant and the facilitator may feel the workshop as wasted effort, affecting how well they may continue on their work after the experience.

Quality and rationale of user stories can vary depending on who is describing them. User stories require good written skills to convey the information behind the user requirement and possible business value. Depending on the background and the current role of the person creating the user stories, the stories may vary heavily. The author had similar results from user story crafting workshops between participants who had a similar or same work position. But with someone else, e.g. a person in management, business, or design, the user stories were different. Thus, for hosting user story crafting workshops, it is important to consider recruiting

such persons who come from such a background that support the motivation of user story crafting. E.g. for systems improvement for an old system, involving an UX or UI designer and an outside user that belongs to the planned user group can give a good contrast of what could be important from design perspective and what is really important for the actual user.

Crafting scenarios allowed for deeper exploration of the why and how areas of user understanding when compared to semi-structured interviewing. Scenarios were used to create comprehensive descriptions of the product's use in its future state. Participants with market knowledge were invited to craft these scenarios, and those resulted in deep discussion and definition of why some particular current or future feature was important to the user and how they would use the feature in some specified context of use. While the description of the actor in scenarios is essential, the reasoning for important actions and descriptions of how they are achieved took precedence in creating the scenarios for the participants. Crafting the scenarios provided deeper reflection than in interviews as the participant had a chance to refine the story they had written as it progressed.

With proper facilitation, scenarios can be helpful in describing the why and how of user understanding, when the facilitator helps the scenario crafter to create such content that aligns with the focus areas of user understanding. Success of creating interesting scenarios from user understanding viewpoint is dependent on the written skills of the person crafting them and the strength of facilitation. Similarly in user stories, scenarios can suffer from having a person who is not the proper representative for the needed user understanding study may not be able to contribute. Written skills become more important in scenarios than user stories, as a whole description of some event requires more focus on keeping the story together and maintaining that red line for the reader of the scenario to understand the big picture. Thus, strength of the facilitation should be considered when hosting scenario crafting workshops.

Scenarios can help with promoting deep discussion and reflection of a topic. Scenarios attempt to describe certain actors in some real world events (Carroll and Rosson, 1990; Carroll, 1999). Asking another person to describe a such story makes them think and ponder of different possibilities, and choose one scenario they want to depict. Depending on the person's imagination and rationale, those scenarios can very well reflecting an actual event. The author had great experiences with crafting those scenarios with another person, as the scenarios gave a couple of good challenges. **First**, crafting scenarios could be used to challenge a claimed expertise. E.g. a project manager may say that they know what is best for the project. But when asked to justify decisions through making scenarios, it may show if some aspects or details were not truly thought thoroughly. **Second**, crafting scenarios can give a glimpse of the person's world view, how they see things happening realistically in their opinion. Understanding the other person's world view can help in facilitating scenario crafting sessions. And **third**, scenarios can promote deep conversation of a topic, such as in interviews. Using good probing questions or simply asking why, the scenario can expand into very detailed and content rich stories.

Scenarios can be used to benchmark customer's ideas and compare current vision with past. With one scenario crafting workshop, the author discussed with one participant in feedback discussion about utilizing scenarios in customer work. They said that scenarios can be a great design practice to help customers to create their vision of their product in such a way that it is understood with common sense and could guide software development work. Scenarios that are created at the beginning of a project and at some time later can be compared to see if there have been any changes in business requirements or other requirements that may affect the software development focus. On a downside, scenarios can be difficult to construct for some who have difficulties with written skills, to convey their ideas and descriptions in a readable and easily understood form.

Prototyping can be used with other design practices. The author had used prototyping as a form of testing and reviewing in semi-structured interview sessions, user story crafting, and scenario crafting workshop sessions. Prototyping can be useful when discussing about some aspects of a product, as humans have various mental models and there cannot be any guarantees that when you describe something, the other person pictures your description exactly in the same way in their head as you do. Thus, showing a prototype can help in discussion revolving around a visual medium, that keeps the discussion more rooted in topics around the prototype. Although, it is important to plan ahead how the prototypes or prototyping is used with other design practices. In interviewing, prototyping can be used in the beginning of the interview as an introductory activity if the interviewee is required to know about the prototype before interviewing. But, if the interviewee needs to be creative and suggest new features and functions, a prototype in the beginning may take away some of that creativity, especially if the prototype is of high fidelity.

Prototyping helped in creating understanding of the current state of the product, identifying limitations, and ideating new features. Prototyping was used to show the current state of the product for the participants, who then had a chance to test the product themselves or ask questions about the product from the author. Prototyping allowed exploring the current functionalities and features to capture a somewhat holistic overview. It also provided a possibility to see the current limitations and shortcomings. Often, after the participants had experienced themselves a limitation in the product, they would offer an idea how that limitation could be solved and what could be the motivation for the user to use the product in the way they propose. This part of prototyping offered answers to the how the product would be used and why it would be used.

Overall, prototyping provides great results for understanding motivations and behaviours in the use of a product. Prototyping in the study allowed the author to show working implementation of the product and then allowing the participant to test the implementation. When the participants were testing the product, this opened a chance for asking basic questions, such as why the person did the sequences of actions in that order or would the person do something differently if they could? If real implementation is not available, paper prototypes or creating wire frame presentations

with PowerPoint or drawing digital images can serve as a substitute. Though, usage of prototypes should be handled with care. Sometimes creativity can be limited when a person sees a high-fidelity prototype in front of them. Finished-looking design may limit how well a person can think outside of the box if they do not have design oriented background or are limited in that area.

Showing static images of the user interface in interviews helped in finding problems and improvements, also as a form of prototyping. The interviews began with a light conversation about the interviewee's perceived important features and functionalities of the product in its current and future state. After these discussions, showing static images of the product's admin panel user interface and asking them to conduct some simple actions through the images helped in finding problems and improvements. In contrast to traditional conversations, reviewing an image was a refreshing and welcome activity in the interviews. Testing ideas and prototyping them by picturing sequences in head seemed to generate more openly ideas than testing the real product in workshops. When eliciting motivational and behavioural aspects of systems improvement (e.g. finding new features or evaluating current ones), the author believes that evaluating static images and inquiring perceived outcomes from conducting some actions in the static pictures could generate more ideas than trying an actual working implementation. Working implementations may set limitations to creativity, if the presented user interface is taken for granted and the person evaluating may not know of better alternatives, as seeing a real implementation may put pressure for the person to propose ideas that should match with the current design. Combining both approaches, and selecting the evaluation of static images to be done before evaluating the real user interface, is what the author did in this thesis, and that may have improved the amount of found problems and suggested improvements found in the study.

Conducting the design practices 1-to-1 could help in making the conversations relaxed. The participants in the sessions were internal users and knew the author. Thus, conducting the sessions 1-to-1 helped in keeping the sessions private and relaxed, allowing the participants to express themselves without fear of creating political or social conflict. However, it is possible that despite the author acting as a researcher and a neutral party during the sessions, there could have been political politeness from the participants, as the product which was evaluated was a shared project between the author's organization and the participants' organization. But, it can be generalized that having 1-to-1 sessions made the event more private and had there been more than one participant within the sessions, the participants may have affected their answers. E.g. pairing a very talkative person and a more silent person may result in the talkative person taking away room from the silent person in answering questions. Also, if a manager and an employee were paired, there could be political pressure for answering in a certain way, e.g. giving such answers that sound beneficial to the product and avoiding suggestions or comments that are not in line with the product strategy or the opinion of the board committee.

User understanding studies through the design practices can strengthen the bonds between internal stakeholders and the product. The study was designed to involve the client and their users in evaluating the product to deepen the current user understanding in the areas of why and how the product is being used. After the interview and workshop sessions, the events were evaluated. In the evaluation, many of the participants had expressed positive feedback for the sessions and design practices. From their feedback, the sessions made the participants more involved with the product than what they normally were, and that the feeling of contribution towards the product's development made them more interested in seeing the product evolve. This lesson was learned with internal users partaking in the study. The study cannot tell whether external users, recruited outside from the internal organization, would show similar results.

Using design practices multiple times a day in sessions can be cognitively exhausting. The design practices described in this thesis require more cognitive abilities than motor skills. Thus, the researcher should note how long, how many, and how frequent sessions they are planning to conduct to a single person. The author had positive feedback for an hour long sessions. In planning the frequency of sessions, the author does not suggest having multiple sessions on the same day for the same participant, as that may be exhausting for them. Choosing the time for the session may also affect, how well they can answer the questions or how creative they can be in the session, e.g. having a session in the Friday evening may not produce same quality of answers as in the morning for the same day. And, having knowledge of the participant's schedule may mitigate the risk of setting a workshop time in between stressful events of the participant. Furthermore, having too many sessions a day, even for different participants, can become too exhausting for the researcher and they should know their limits, when scheduling the events.

Small companies can take advantage of insider action research. Both the case company and the client company are small companies, with less than or equal to 30 employees in both. When asked from them about the costs of this insider action research, they did not see the costs of used effort as high, compared to the results it gave them. It is therefore possible to have insider action research as part of active software development effort, as there could not be a need to weigh between doing insider action research or active product development. Had the study failed, a four days worth of work may have not set back the project in this case.

Doing insider action research may provide great results to system improvement. The client company, who ordered this insider action research, gained a good amount of results that could guide further software development efforts. Effort wise, the author used just about four full workdays worth of hours to provide meaningful results. While not every insider action research cycle can be predicted as successful, this study presented good results to the client company, which gives empirical evidence of it supporting a systems improvement project, with certain requirements fulfilled, such as having a defined motivation and set goals for the study. As the author had been part of the system improvement process and in a management

role, he had a great opportunity to assist the improvement process in assisting each participant to form improvement ideas in the design workshops. However, potential bias and danger may come from pre-knowledge and role duality, as the author may have affected the decisions of some participants due to leading questions or conversation due to pre-knowledge. Despite the author attempting to stay neutral, there is a possibility for unconscious actions that are realized afterwards to have caused bias to the participants' answers.

For new consultancy practitioners, insider action research can help in increasing user understanding in software development projects. The author worked, at the time of this thesis, as a consultant for the client company. His role as a project manager made him part of the product's development team and thus an insider to the project he was working on, according to the insider action research definition (Coghlan and Holian, 2007). The empirical study was first of its kind to the author, and the insider action research process provided a great foundation to start with for the author. As more similar projects come across, the process will evolve, as in agile software projects through inspection and adaption as in e.g. Scrum (Deemer et al., 2012). Until the author, as a new practitioner in the field of user understanding studies, gains enough knowledge and experience, he continues using insider action research as a template until it evolves or a better template is found.

General experiences for the author from conducting the study. The author had not conducted insider action research studies before. In total, the insider action research took 30 hours and 45 minutes. The whole study spanned around a 5 week period, giving an approximate of 6 hours and 09 minutes on average of work in a week. In Scrum, an ideal maximum length of a sprint is 4 weeks (Deemer et al., 2012), therefore, if the author had managed to stay in 4 weeks, it could be said that the insider action research could fit Scrum like sprint based agile development cycles. Despite the mismatch with sprint length, the author believes that the insider action research, as he did it, can fit as it into the case company agile work process, which takes after Scrum, modified to the company needs.

In reflection to other software projects done in the case company, an insider action research study as how it was described in this thesis, could fit alongside the software development projects. If we consider the total effort used (30 hours 45 minutes = about 4 days of work), a better planned scheduling may have allowed a similar study to be done in less than 2 weeks, given that recruited individuals have good availability to participate in the studies and the amount of recruited people stays the same. Overall, the resulted total effort (of about 4 days) does fit within the case company software development cycles, allowing for future studies to be done alongside software development. The user understanding was increased in the case project and the results for the client were satisfactory.

The author was confident in conducting more insider action research studies in the future. Challenges may arise when some other design practices are needed, which are not described in this thesis. These challenges include using time for self study

and testing the design practices before using them on a real customer case. However, this is a natural challenge in any field of work, where one cannot remain rooted in a set of knowledge or skills, but must expand those to maintain the ability to tackle modern problems in our ever changing world driven by digitalization.

5 Discussion

5.1 RQ1: Which design practices fit to the case company context?

Semi-structured interviews, task analysis, user stories, scenarios, and prototyping were chosen for analysis. These design practices were chosen based on an interview with the thesis advisors to find five most promising sounding design practices. The literature review took into account all five design practices. The empirical study, however, excluded task analysis as it did not fit into the study's scope. Therefore, the empirical study has evaluated only semi-structured interviews, user stories, scenarios, and prototyping.

The four design practices were evaluated in the empirical study against software development agility and the satisfaction of user understanding increment in the case project. The total effort of about 4 days of work (30 hours and 45 minutes) were seen fitting into the case company's agile software development process. The satisfaction of the client for the user understanding increasing results gave assurance for accepting the four design practices into the case company's context.

Semi-structured interviews were used to elicit important features for the present and future versions of the product. The used interview template allowed for eliciting important features for the users. The importance was reasoned with identifying relevant values and motivations for each feature. This was supported by a small prototyping session during the interview. Prototyping in the interview consisted of viewing paper printed images of the current UI, to find improvements or to highlight problems in it.

Using the semi-structured interviews and prototyping together supported agile software development. Continuous delivery of software was supported by eliciting system improvements. Working software was ensured by highlighting problems in the current version of the product. Software quality was assisted by forming a list of found problems for bug fixing prioritization. Customer collaboration was ensured by having the product owner participate in the process.

User stories were used to elicit user requirements for the product, with the focus of understanding motivations for the use. User stories described user requirements in the users own words. Forming of the user stories was guided by asking for motivators for the described requirements, emphasizing the value described in the end of a user story.

User stories supported continuous software delivery by providing descriptions of new features that are important to the users. User collaboration was supported by partaking users in forming the user stories. Customer collaboration was present by allowing the product owner to evaluate the created stories and choose those for further development. Software development could be guided by user stories, where

a quality user story alone could explain the feature in such detail that a developer could understand the motivation and value behind it.

Scenarios were used to create whole descriptions of real world events. Scenarios elicited user requirements on a larger scale than the user stories could describe. Scenarios provided detailed descriptions of the users, their interactions, circumstances, goals, and environment.

Scenarios supported continuous software delivery by providing some large concepts, that describe a set of features happening in some context. Working software was supported by the detailed descriptions from scenarios that explain well a function of a sub-system in the product. Quality of work was taken into account by following the requirements set by the scenarios, given that the scenarios did not create conflicts. Customer collaboration was supported by engaging the product owner and the project manager in creating, analyzing, and comparing the scenarios.

Prototyping was used with other design practices to support understanding of the product. Prototyping was used as a way to test the current version of the product, and to find possible design problems. Working software and quality of work was supported by finding bugs in the product in real context. All study participants partook in testing the product.

Task analysis was not used in the empirical study, as it did not fit the scope. Task analysis can be a rigorous analysis that looks into each step taken, and beyond, to achieve the desired goal (Courage et al., 2008; Norman, 1988, 2013). Therefore, task analysis can take a considerable amount of time and the value of its results should be well analyzed before committing to using it. Although, Courage et al. (2008) state that task analysis can take an iterative approach. The practitioner is responsible for specifying the length of an iteration and its contents, so that the scope, focus, and granularity won't exceed that which was planned.

In the case project, the product owner had left task analysis outside of the scope. Without empirical proof for using task analysis in a case project, the author must exclude task analysis from the list of design practices that fit to the case company context. The literature review gives details to test task analysis for discovering user understanding in the future.

These results are based on a single cycle of insider action research. The listed results give a positive image of the four design practices fitting into the case company context. The results did not highlight possible challenges, such as the study participants being unavailable or the necessity of time to travel to a workshop location. More iterative cycles would be recommended to provide more proof for the design practices to fit the case company context in more than one cycle.

5.2 RQ2: How do the selected design practices work?

The background for the functionality of the design practices was studied through literature review. To prepare the author for conducting any of the studied design practices, it was imperative to understand through literature the fundamentals for each five design practices researched in this thesis.

For semi-structured interviews, 9 publications were reviewed to describe how the design practice works (Adams, 2010; Beyer and Holtzblatt, 1998; Bryman, 2016; Hackos and Redish, 1998; Kujala and Kauppinen, 2004; Kvale, 1996, 2007; Rabionet, 2011; Wilson, 2013). According to Adams (2010), Beyer and Holtzblatt (1998), Hackos and Redish (1998), Bryman (2016), and Wilson (2013), semi-structured interviews consist of three sections: (1) preparation for the interview, (2) conducting the interview, and (3) what to do after the interview.

Semi-structured interviews are prepared by forming the interview goal or research purpose (Wilson, 2013). Those will guide the forming of interview questions, which include general, specific questions, and *ad-hoc* questions (Hackos and Redish, 1998; Wilson, 2013). An interview script greatly assists in conducting the interviews and staying in the interview topics (Hackos and Redish, 1998; Kvale, 2007; Wilson, 2013). One could consider "interviewing as many participant as necessary" (Kvale, 1996), but should take into consideration the available time and resources (Kujala and Kauppinen, 2004). 5 or 6 to 10 users can produce a good breadth of results (Beyer and Holtzblatt, 1998; Hackos and Redish, 1998).

When conducting interviews, the interviewer should act calm, polite, and professional with emotional control (Adams, 2010; Wilson, 2013). The interviewee must know about the purpose of the interview, length, arrangements, how their data is handled (Wilson, 2013), and when the interview has ended (Adams, 2010). The interviewer should not be afraid of silence. Instead, give room for reflection through silence when encountering it during an interview (Kvale, 2007).

After the interview, it is suggested to finalize the interview notes from fresh memory if possible (Adams, 2010; Wilson, 2013). Enough time should be booked for analyzing the interview data (Wilson, 2013). Audio recordings can be transcribed with focusing only on the interesting parts of the interview to optimize time usage (Bryman, 2016).

Task analysis focuses on discovering users' actions, goals, and capturing their real objectives through a real demonstration of their work (Hackos and Redish, 1998). It utilizes different practices to achieve this: interviewing users, observing users, shadowing users, and conducting day in life studies (Hackos and Redish, 1998; Courage et al., 2008). Courage et al. (2008) set four principles for task analysis. First, "task analysis is an integral part of a broader analysis", meaning it alone cannot contribute enough for designing or evaluating a product (Courage et al., 2008). Second, "task analysis is about understanding the user's goals" (Courage et al., 2008). E.g. the analysis could be based on Norman's (2013) 7-stage of action model.

Third, "task analysis is relevant through all stages of the process", where it belongs to planning, designing, developing, and evaluating a product (Courage et al., 2008). And finally, fourth, "practical reality impinges on what we actually do", meaning considerations for real issues such as time, resources, people, travel restrictions, as well as marking the difference of analysis between market research (breadth) and task analysis (depth) (Courage et al., 2008).

User stories follow the format of: "I as a (role) want (function) so that (business value)" (Cohn, 2004). User stories are clear sentences that describe *the role*, *the goal or the task*, and *the business value* (Cohn, 2004; Deemer et al., 2012; Lucassen et al., 2016). The role defines the single actor (the user), who belongs to an identified user group that the story addresses (Cohn, 2004; Lucassen et al., 2016). The goal or the task of the user story describes what the user tries to achieve through an active voice with relevant action verbs (want, am able, feel, browse) (Cohn, 2004; Lucassen et al., 2016). The business value describes the motivation or feelings the user undergoes in the user story (Cohn, 2004). It can also provide clarification to the goal or the task, describe dependencies to another functionality, or describe a quality requirement (Lucassen et al., 2016).

Scenarios are written stories that describe in detail how users may interact with a product with a set of goals and tasks, and within a specific environment (Carroll and Rosson, 1990; Carroll, 1999). The scenarios speak the users' language, so that non-technical people can understand them. simple language allows for finding defects in the scenario, such as errors and omissions (Alexander and Maiden, 2004). Writing scenarios can be an iterative process, where first the writer crafts the initial, rough version (Carroll, 1999), and then refine the story by including details (Carroll, 2002). Open ended scenarios can be constructed so that some key-details are omitted on purpose. These are made by leaving e.g. the proposed solution out on purpose to allow creative solution thinking (Bødker, 1999). In contrast, closed scenarios describe the proposed solution in high detail from start to finish, which allows for using the scenario as a testing guide (Bødker, 1999).

Prototyping allows for a prototype of a system to be used for: understanding the problem context, finding alternative solutions, understanding product value, understanding user flows, and using the prototype as a tool for communication (McElroy, 2016). Prototyping can be a collection of activities to flesh out the prototype, such as sketching and modelling, and then presenting the prototype for critique for improvements or testing the prototype (McElroy, 2016; Warfell, 2009). Prototyping provides various uses for refining, evolving, and improving ideas and concepts into practical prototypes. Fidelity of a prototype should be taken into account when planning the time to build the prototype, so that the appropriate amount of fidelity matches with the expected value gained from the prototype (McElroy, 2016).

5.3 RQ3: How can the design practices increase user understanding?

User understanding was defined through inspection of three main areas. **Who uses**, which answers to the relevant user groups. **Why they use** or **why won't they use**, which answers to the motivational and emotional aspects of usage. **How they use**, which answers to how the product is used and how well is the product designed for its users.

Semi-structured interviews were conducted to explore values and motivations for the product's use by focusing on the why and how the product can be used in its current and future versions. Semi-structured interviews were planned to elicit important features and functionalities for the product's admin panel in its current and future versions. The questions focused on finding answers mainly to the motivational and emotional aspects of user understanding (why they use), attempting to elicit reasoning for the value the product creates or could create, and why the interviewee sees some features and functionalities important for the inspected product. The outcomes from the interviews were deemed helpful by the client who ordered the study and had described well rationale for importance, motivational, and emotional aspects of the inspected product for its current and future state. Some probing questions focused on eliciting behavioural aspects of the product use.

User stories were crafted to find user requirements for the product with various participants by focusing in understanding who the users are and why they would use the product. The crafting of user stories resulted in a total of 67 user stories, which described the actor, the requirement, and sometimes the motivation or gained value in each story. Therefore, user stories contributed towards understanding the relevant user group, to which the described actor in the story belonged to (who uses), and the motivational, emotional, or value-driven reason for the requirement (why they use).

Scenarios were used to describe real world events that were relevant for improving the product by understanding who was doing what for which reason and how. Scenarios were used to elicit mainly the why and how of the user understanding with short descriptions of the actors (who). With the focus for improving the inspected product's admin panel, scenario crafters made whole descriptions of certain events, that depicted the actors, the sequences of actions, their rationales, motivations, emotions, values, circumstances and behaviour in the scenarios. Each crafted scenario contained differing amount of detail, which needs to be taken into account when comparing the created user understanding between scenarios.

Prototyping was used together with other design practices to study the why and how questions of user understanding. Prototyping the product provided valuable information about the motivations and emotions for using the product, and how the product would be used. This was achieved by interviewing and observing the person testing the product. Prototyping gave most honest information

about the product use, when the person prototyping was asked to do some tasks with the product. Their actions, body language, and behaviour in the product's UI gave more valuable information on how they used the product than how they may have verbally described their use. Prototyping contributed in the why and how areas of user understanding, with focus on the latter.

Task analysis contributes to user understanding in the why and how questions of user understanding, based on literature. Because task analysis was excluded in the empirical study, its contribution to user understanding is reviewed through literature. Task analysis focuses in understanding the user and their context of use. This includes understanding the users, their tasks and goals, and the environment where the usage occurs (Hackos and Redish, 1998; Courage et al., 2008). The 7-stage of action model by Norman (2013) explains how reaching the goal through actions allows for an evaluation of the motivations for such action (why they use) and the performance itself (how they use). While understanding the actor is important, emphasis for user understanding in task analysis is in understanding why and how the user uses the product.

5.4 Limitations of the study

Should the insider action researcher be as honest as possible? The insider action researcher should be wary about the meaning of honesty. Not meaning that the researcher should lie, however, the researcher should think through how honesty is brought up to the client. The author knew well the client he was working with, but same approach may have not worked with other clients. Being too honest to a client that is sensitive to the product may create difficult social situations, that may affect the chances of continuing the study after the first cycle or even affect the partnership between the case and client company. Thus, the author suggests for insider action researchers to be as politely honest as possible, attempting to not create social conflict and negative attitudes, and at the same time give as honest information as they can within the said limitations. While the author had not yet experienced such situations with insider action research, he has witnessed in another case his honesty without politeness resulting in misunderstood attempt to assist a case as a pessimistic review of a product.

The challenge of the "insider" role hindering the neutrality of the researcher. Hosting sessions with a familiar person may cause the researcher to slip with neutrality and to forget to take enough distance from the situation to remain as an observer. The author at times forgot his role as a neutral researcher and skipped some steps in the interview sessions or workshop sessions. This was depending on whether the author knew for sure if the other person knew e.g. how to create scenarios and skipped the proper introduction. Or, the author may have used leading questions. By not following the planned procedure and taking distance from the participant, some valuable information may have been lost or biased, because the researcher assumed the other to know the answer or answer in some particular way.

The insider action research could have gained more with having participants come outside of the case company or the client company. Both of the companies shared the project and had been developing the product forward. But, 3 out of 5 participants had not been in touch with the product enough to claim knowing the functions and features of the product. But, internal participants may provide biased answers due to multiple possible reasons: (1) They want to avoid internal conflicts, e.g. they would say something that the author could have not acted neutral towards and cause conflict, (2) They act as a "friendly test subject", wanting to provide results they think the researcher wants to hear, so that their research could go forward, (3) Political pressure, e.g. thinking that criticism towards company's own product will hinder the person's position in the company, affect career advancements, or create social conflicts (example similar to Coghlan and Holian's (2007) notes about failed insider action study and consequences to the researcher). Thus, the author believes that involving "neutral 3rd parties" could provide less biased answers and in case of conflicts, there should not be any internal conflicts caused by 3rd party comments.

The results cannot be generalized. The empirical study was conducted to a client product, which represents a unique set of stakeholders and a unique product. Similar products may exist, but are not exactly the same. Therefore, the results from the study are qualitative and apply for the specific situation the study happened. Replicating the study for another product in another environment and stakeholders may not yield similar results.

Also, it is possible that the first study cycle may fail. This thesis could not provide an example case where the insider action research could fail and how it would have affected the practitioner, the study, stakeholder organizations, and the continuity of the user understanding study efforts.

6 Conclusions

The goal of this thesis was to understand, which design practices can help small companies to increase their user understanding in agile software development projects. Task analysis was not included in the empirical study, hence its contribution to increasing user understanding in the case company context could not be evaluated.

A small company can use semi-structured interviews, user stories, scenarios, and prototyping to increase user understanding in agile software development projects. The listed design practices had provided user understanding for the inspected product's current and future state. The design practices answered to important questions of user understanding. The design practices answered most to *why the users use the product*, describing the motivations and values for product use. Then, the design practices answered to *how the users use the product*, describing how the usage happened in the product. And lastly, the design practices answered to *who uses the product*, describing the user groups relevant to the product.

Semi-structured interviews with prototyping allowed for open discussions with the participants about the problems in the product. Interview questions explored suggestions for user requirements, motivations, values, and design options for new features. User stories described the users, their needs and wants, and the created value in clear sentences. User stories follow a template, therefore creating them was a lightweight activity. Scenarios described user requirements, motivations, goals, resources, environment and circumstances on a larger scale. Scenarios attempted to describe a big picture for the product use realistically. Prototyping supported other design practices by allowing the users to test the product. Testing the product helped users to give reasons for how the product could be used and why would it be used in their described way.

Conducting the design practices to five participants resulted in meaningful and valuable results to the product owner. Analyzing the interview data, user stories, and scenarios was deemed lightweight and engaging by the product owner. Compatibility with agile software development was seen possible by the case company. The author's effort for the user understanding study had taken a little over 4 days of effort, meaning for the work to easily fit into a 2 to 4 week software development cycle.

Increasing user understanding may need contribution from several design practices. The design practices produced qualitative data, which means that to analyze the data reliably, there should be many data sources. A design practice alone may not be able to contribute to all areas of user understanding. A couple of design practices may not provide enough breadth or depth for the wanted user understanding. Thus, the author suggests using several design practices, that contribute to the overlapping areas of user understanding. This ensures that the user understanding produced by the design practices can be compared against each other for increasing the credibility of analysis. However, not all cases require the use of several design practices. A single design practice can produce meaningful results.

Insider action research can help new consulting practitioners to conduct user understanding studies. For new practitioners for conducting user understanding studies (which the author was as well), insider action research can provide supporting framework. Being involved in design activities for a product may often mean active participation in the product's development. Insider action research provides ideas for taking a neutral stand and planning the use of such design practices that can provide answers to the wanted areas of user understanding.

For the future, using more design practices with insider action research to increase user understanding in agile software development projects would be interesting. This work has given an example of a case project with four design practices. A single cycle of user understanding study, as described in this thesis, is not enough to say whether this concept could produce similar or even better results in future iterations. Therefore, more iterations with the same framework is needed to bring credibility for this type of user understanding increasing study.

It would be interesting to know whether this concept can be used in other cases. For example, could large companies use this concept to increase user understanding in their agile software development projects? How would a different amount of users affect the data from the selected design practices? How would a different combination of design practices affect the study? For companies that want to increase user understanding in their agile software development projects, they could try this concept and modify it to fit their context as necessary.

References

- Adams, E. (2010), ‘The joys and challenges of semi-structured interviewing’, *Community Practitioner* **83**(7), pp. 18 – 21. [online] Available at: <https://search.proquest.com/docview/754140856?accountid=27468> [Accessed 17. Jul. 2019].
- Ahmed, S., McKnight, C. and Oppenheim, C. (2006), ‘A user-centred design and evaluation of IR interfaces’, *Journal of Librarianship and Information Science* **38**(3), 157–172. [online] Available at: <https://journals.sagepub.com/doi/abs/10.1177/0961000606063882> [Accessed 04. Oct. 2019].
- Alexander, I. and Maiden, N. (2004), ‘Scenarios, stories, and use cases: The modern basis for system development’, *Computing and Control Engineering* **15**(5), pp. 24–29. [online] Available at: https://digital-library.theiet.org/content/journals/10.1049/cce_20040505 [Accessed 03. Jul. 2019].
- Almquist, E., Senior, J. and Bloch, N. (2016), ‘The elements of value’, *Harvard Business Review* pp. 47–53. [online] Available at: <https://hbr.org/2016/09/the-elements-of-value> [Accessed 12. Sep. 2019].
- Annett, J. (2003), Hierarchical task analysis, in E. Hollnagel, ed., ‘Handbook of Cognitive Task Design’, New Jersey: Lawrence Erlbaum Associates, Inc, pp. 17 – 35. [online] Available at: https://mycourses.aalto.fi/pluginfile.php/614499/mod_resource/content/1/Annett_2003.pdf [Accessed 29. Jun. 2019].
- Avison, D., Lau, F., Myers, M. and Nielsen, P. (1999), ‘Action research’, *Communications of the ACM* **42**(1), 94 – 97. [online] Available at: <https://dl.acm.org/citation.cfm?id=291479> [Accessed 17. Oct. 2019].
- Bardram, J. (2000), ‘Scenario-based design of cooperative systems: Re-designing an hospital information system in denmark’, *Group Decision and Negotiation* **9**(3), pp. 237–250. [online] Available at: <https://link.springer.com/article/10.1023/A:1008743509338#Bib1> [Accessed 03. Oct. 2019].
- Beaudouin-Lafon, M. and Mackay, W. (2008), Prototyping tools and techniques, in A. Sears and J. Jacko, eds, ‘The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications’, New York, NY, USA: Taylor & Francis Group, pp. 927 – 947.
- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, M., Sutherland, J. and Thomas, D. (2001), ‘Manifesto for agile software development’. [online] Manifesto for Agile Software Development. Available at: <https://agilemanifesto.org/> [Accessed 31. Feb. 2020].
- Bevan, N. (2009), Criteria for selecting methods in user centred design, in ‘Conference: I-USED’09: The Interplay between Usability Evaluation and Software Development. Proceedings of the Second International Workshop on the Interplay between Usability Evaluation and Software Development,

- Uppsala, Sweden, August 24, 2009', ResearchGate. [online] Available at: https://www.researchgate.net/publication/221508484_Criteria_for_Selecting_Methods_in_User_Centred_Design [Accessed 28. Oct. 2019].
- Beyer, H. and Holtzblatt, K. (1998), *Contextual Design*, San Francisco, CA, USA: Academic Press.
- Brhel, M., Meth, H., Maedche, A. and Werder, K. (2015), 'Exploring principles of user-centered agile software development: A literature review', *Information and Software Technology* **61**, 163–181. [online] Available at: <https://doi.org/10.1016/j.infsof.2015.01.004> [Accessed 10. Feb. 2020].
- Bryman, A. (2016), *Social Research Methods*, 5th edn, New York, NY, USA: Oxford University Press.
- Buchenau, M. and Suri, J. (2000), Experience prototyping, in 'DIS '00', ACM, pp. pp. 424 – 433. [online] Available at: <http://designblog.uniandes.edu.co/blogs/dise2100/files/2010/08/Experience-Prototyping.pdf> [Accessed 19. Aug. 2019].
- Bødker, S. (1999), Scenarios in user-centred design - setting the stage for reflection and action, in 'Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences', IEEE. [online] Available at: <https://ieeexplore.ieee.org/abstract/document/772892> [Accessed 05. Jul. 2019].
- Camburn, B., Viswanathan, V., Linsey, J., Anderson, D., Jensen, D., Crawford, R., Otto, K. and Wood, K. (2017), 'Design prototyping methods: state of the art in strategies, techniques, and guidelines', *Design Science* **3**(e13), pp. 33. [online] Available at: <https://doi.org/10.1017/S1431927611000523> [Accessed 10. Aug. 2019].
- Carroll, J. (1999), Five reasons for scenario-based design, in 'Proceedings of the 32nd Hawaii International Conference on System Sciences', IEEE. [online] Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.459.7467&rep=rep1&type=pdf> [Accessed 04. Jul. 2019].
- Carroll, J. (2002), Scenarios and design cognition, in 'Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02)', IEEE. [online] Available at: <https://ieeexplore.ieee.org/document/1048498> [Accessed 30. Sep. 2019].
- Carroll, J. and Rosson, M. (1990), Human-computer interaction scenarios as a design representation, in 'Twenty-Third Annual Hawaii International Conference on System Sciences', IEEE. [online] Available at: <https://ieeexplore.ieee.org/document/205231> [Accessed 01. Oct. 2019].
- Cerejo, L. (2018), *How To Get To Know Your Users*, Smashing Magazine. [online] Available at: <https://www.smashingmagazine.com/2018/06/how-to-get-to-know-your-users/> [Accessed 10. Sep. 2019].

- Coghlan, D. (2001), 'Insider action research projects: Implications for practicing managers', *Management Learning* **32**(1), 49 – 60. [online] Available at: <https://journals.sagepub.com/doi/abs/10.1177/1350507601321004> [Accessed 17. Oct. 2019].
- Coghlan, D. and Brannick, T. (2005), *Doing Action Research In Your Own Organization*, 2nd edn, London: Sage Publications Ltd. [online] Available at: <http://213.55.83.214:8181/Education/27492.pdf> [Accessed 17. Oct. 2019].
- Coghlan, D. and Holian, R. (2007), 'Editorial: insider action research', *Action Research* **5**(1), 5 – 10. [online] Available at: <https://journals.sagepub.com/doi/10.1177/1476750307072872> [Accessed 18. Oct. 2019].
- Cohn, M. (2004), 'User stories applied: For agile software development'. [ebook] Boston: Pearson Education, Inc. Available at: <https://learning.oreilly.com/library/view/user-stories-applied/0321205685/?ar> [Accessed 10. Jul. 2019].
- Courage, C., Redish, J. and D., W. (2008), Task analysis, in A. Sears and J. Jacko, eds, 'The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications', New York, NY, USA: Taylor & Francis Group, pp. 927 – 947.
- Cross, N. (2001), Design cognition: Results from protocol and other empirical studies of design activity, in M. M. C. Eastman and W. Newstetters, eds, 'Design knowing and learning: Cognition in design education', Elsevier. [online] Available at: <https://www.sciencedirect.com/science/article/pii/B978008043868950005X?via%3Dihub> [Accessed 20. Sep. 2019].
- Crystal, A. and Ellington, B. (2004), Task analysis and human-computer interaction: approaches, techniques, and levels of analysis, in 'AMCIS 2004 Proceedings', pp. 3202–3210. [online] Available at: <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1967&context=amcis2004> [Accessed 03. Jul. 2019].
- Deemer, P., Benefield, G., Larman, C. and B., V. (2012), *A Lightweight Guide to the Theory and Practice of Scrum - Version 2.0*, ScrumPrimer. [online] Available at: https://scrumprimer.org/scrumprimer20_small.pdf [Accessed 11. Oct. 2019].
- Faulkner, X. (2000), *Usability Engineering*, New York: PALGRAVE.
- Gabbard, J., Hix, D. and Swan, J. (1999), 'User-centered design and evaluation of virtual environments', *IEEE Computer Graphics and Applications* **19**(6), 51 – 59. [online] Available at: <https://doi.org/10.1109/38.799740> [Accessed 04. Oct. 2019].
- Gautam, S. (2012), *Emotions and Motivations*, Psychology Today. [online] Available at: <https://www.psychologytoday.com/intl/blog/the-fundamental-four/201205/emotions-and-motivations> [Accessed 13. Sep. 2019].

- Hackos, J. and Redish, J. (1998), *User and Task Analysis for Interface Design*, New York, NY, USA: John Wiley & Sons, Inc.
- Holzinger, A. (2005), ‘Usability engineering methods for software developers’, *Communications of the ACM* **48**(1), pp. 71 – 74. [online] Available at: https://www.researchgate.net/publication/220422205_Usability_Engineering_Methods_For_Software_Developers [Accessed 01. Aug. 2019].
- Hornsby, P. (2010), ‘Hierarchical task analysis’. [online] UXmatters. Available at: <https://www.uxmatters.com/mt/archives/2010/02/hierarchical-task-analysis.php> [Accessed 26. Jun. 2019].
- Interaction-design.org (2019), ‘Design scenarios - communicating the small steps in the user experience | interaction design foundation’. [online] Available at: <https://www.interaction-design.org/literature/article/design-scenarios-communicating-the-small-steps-in-the-user-experience> [Accessed 04. Jul. 2019].
- Kauppinen, M., Savolainen, J., Lehtola, L., Komssi, M., Töhönen, H. and Davis, A. (2009), From feature development to customer value creation, in ‘17th IEEE International Requirements Engineering Conference’, Atlanta, USA: IEEE, pp. 275–280. [online] Available at: <https://ieeexplore.ieee.org/document/5328526> [Accessed 24. Sep. 2019].
- Kujala, S. and Kauppinen, M. (2004), Identifying and selecting users for user-centered design, in ‘NordiCHI ’04’, New York, NY, USA: ACM, pp. 297 – 303. [online] Available at: <https://dl.acm.org/citation.cfm?id=1028060> [Accessed 17. Jul. 2019].
- Kvale, S. (1996), *Interviews: An Introduction to Qualitative Research Interviewing*, California, CA, USA: Sage Publications, Inc.
- Kvale, S. (2007), *Doing Interviews*, London, UK: SAGE Publications Ltd. [ebook] Available at: <https://methods.sagepub.com/book/doing-interviews> [Accessed 22. Jul. 2019].
- Lucassen, G., Dalpiaz, F., van der Werf, J. and Brinkkemper, S. (2016), ‘Improving agile requirements: the quality user story framework and tool’, *Requirements Engineering* **21**(3). [online] Available at: <https://link.springer.com/article/10.1007/s00766-016-0250-x#enumeration> [Accessed 10. Jul. 2019].
- Maslow, A. (1943), ‘A theory of human motivation’, *Psychological Review* **50**(4), pp. 370 – 396. [online] Available at: <https://psycnet.apa.org/doiLanding?doi=10.1037%2Fh0054346> [Accessed 13. Sep. 2019].
- McElroy, K. (2016), *Prototyping for Designers*, O’Reilly Media, Inc. [ebook] Available at: <https://learning.oreilly.com/library/view/prototyping-for-designers/9781491954072/> [Accessed 19. Aug. 2019].
- McLeod, S. (2018), *Maslow’s Hierarchy of Needs*, SimplyPsychology. [online]

- Available at: <https://www.simplypsychology.org/maslow.html> [Accessed 13. Sep. 2019].
- McNiff, J. (2013), *Action Research: Principles and Practice*, 3rd edn, London: Routledge. [online] Available at: <https://www.taylorfrancis.com/books/9780203112755> [Accessed 17. Oct. 2019].
- Nielsen, J. (1993), *Usability Engineering*, San Diego: Academic Press.
- Norman, D. (1988), *The Psychology of Everyday Things*, New York: Basic Books.
- Norman, D. (2013), *The Design of Everyday Things*, New York, NY, USA: Basic Books.
- Patton, M. (2002), *Qualitative Research & Evaluation Methods*, 3rd edn, CA, USA: Sage Publications, Inc.
- Quesenbery, W. (2001), ‘What does usability mean: Looking beyond ‘ease of use’’. [online] WQusability. Available at: <http://www.wqusability.com/articles/more-than-ease-of-use.html> [Accessed 03. Oct. 2019].
- Rabionet, S. (2011), ‘How i learned to design and conduct semi-structured interviews: An ongoing and continuous journey’, *The Qualitative Report* **16**(2), 563 – 566. [online] Available at: <https://nsuworks.nova.edu/tqr/vol16/iss2/13/> [Accessed 17. Jul. 2019].
- Ritter, F., Churchill, E. and Baxter, G. (2014), *Foundations for Designing User-Centered Systems: What System Designers Need to Know about People*, London: Springer. [ebook] Available at: <https://link.springer.com/book/10.1007/978-1-4471-5134-0> [Accessed 29. Jul. 2019].
- Rosson, M. and Carroll, J. (2008), Scenario-based design, in A. Sears and J. Jacko, eds, ‘The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications’, New York, NY, USA: Taylor & Francis Group, pp. 1041 – 1060.
- SFS-EN ISO 9241-11 (2018), *Ergonomics of human-system interaction. Part 11: Usability: Definitions and concepts*, Helsinki: Suomen Standardisoimisliitto SFS ry. [online] Available at: <https://online.sfs.fi/> [Accessed 15. Aug. 2019].
- SFS-EN ISO 9241-210 (2019), *Ergonomics of human-system interaction. Part 210: Human-centred design for interactive systems*, Helsinki: Suomen Standardisoimisliitto SFS ry. [online] Available at: <https://online.sfs.fi/> [Accessed 15. Aug. 2019].
- Stanton, N. (2006), ‘Hierarchical task analysis: Developments, applications, and extensions’, *Applied Ergonomics* **37**(1), pp. 55–79. [online] Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0003687005000980> [Accessed 03. Jul. 2019].

- Thornton, P. (2019), *Go beyond user interviews with contextual inquiry*, UX Collective. [online] Available at: <https://uxdesign.cc/contextual-inquiry-a-primer-14e2e0696fb9> [Accessed 10. Oct. 2019].
- Vredenburg, K., Mao, J., Smith, P. and Carey, T. (2002), A survey of user-centered design practice, in 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', CHI '02, ACM, New York, NY, USA, pp. pp. 471–478. [online] Available at: <https://doi.org/10.1145/503376.503460> [Accessed 09. Jul. 2019].
- Warfell, T. (2009), *Prototyping*, Rosenfeld Media. [ebook] Available at: <https://learning.oreilly.com/library/view/prototyping/9781457102417/> [Accessed 19. Aug. 2019].
- Wilson, C. (2013), 'Interview techniques for ux practitioners'. [ebook] Waltham, MA, USA: Morgan Kaufmann. Available at: <https://learning.oreilly.com/library/view/interview-techniques-for/9780124103931/> [Accessed 10. Oct. 2019].
- Wixon, D. (2003), 'Evaluating usability methods: why the current literature fails the practitioner', *Interactions - The digital muse: HCI in support of creativity* **10**(4), 28–34.
- Zowghi, D. and Coulin, C. (2005), Requirements elicitation: A survey of techniques, approaches, and tools, in 'Engineering and Managing Software Requirements', Springer, Berlin, Heidelberg, pp. 471–478. [online] Available at: <https://opus.lib.uts.edu.au/bitstream/10453/11626/1/2005003295.pdf> [Accessed 14. Jul. 2019].