Designing AI-Based Systems for Qualitative Data Collection and Analysis

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Tim Rietz, M.Sc.

Tag der mündlichen Prüfung:	01.07.2021
Referent:	Prof. Dr. Alexander Mädche
Korreferent:	Prof. Dr. Paola Spoletini
Karlsruhe	Mai 2021

Acknowledgments

Having started my PhD studies in December 2017, I remember the past three and a half years as a series of ups and downs, which probably goes for everything in life. Looking back at this exciting, inspiring, and challenging time, I distinctly remember many ups, while the downs seem almost forgotten. To a large extent, I attribute this to the wonderful people that I got to meet along the way, who never failed to make my time as a PhD student and as an IT consultant fun.

Certainly, I want to thank my mentor and PhD supervisor Prof. Dr. Alexander Mädche, for his guidance, inspiration, and feedback throughout my studies. While I did not know what to expect when I started my position at the institute, I quickly learned how lucky I was with my choice of a supervisor. Alexander always had an open door for my questions, ideas, and concerns. He also actively seeked updates on my process and encourage me to submit my research to prestigious outlets. I am incredibly grateful for your support. On that note, I also want to thank Prof. Dr. Paola Spoletini, Prof. Dr. Hagen Lindstädt, and Prof. Dr. Wolf Fichtner for taking the time to serve on my PhD committee. I enjoyed discussing my research with you! Special thanks go to Paola, for welcoming me into the Requirements Engineering community on Jeju Island in 2019 and helping me to establish a network in RE!

I want to express my gratitude to my colleagues at the Institute of Information Systems and Marketing (IISM) and at the research group for Information Systems and Service Design (ISSD) in particular. Thank you for always helping with pre-tests, for providing valuable feedback during our research meetings, for the numerous coffee breaks, for laughing together about the weird intricacies of publication processes, and for several highly entertaining social events. Naturally, there are some colleagues whom I had the chance to spend more time with, while I met others only occasionally. Regardless, I always felt that we acted as a team, celebrating successes and learning from failures together. For that, I am grateful.

Besides my research at the IISM, I completed four projects as business analyst as part of the IT consultancy Senacor. In fact, my first workday after finishing my Masters was to receive my Senacor laptop and phone, as well as an introduction at the Senacor headquarters in Nürnberg. Afterwards, I jumped on a train to Frankfurt for the Senacor Christmas Party. Not a bad start! Thank you to all the great colleagues that I had the chance to meet during my introduction to Senacor, the team events, and of course, the projects. Thanks to your support and guidance, I learned much about IT projects, architectures, stakeholder management, and, naturally, great Powerpoint slide design. I am convinced that the insights I gained helped me with speaking the RE language in my research.

I am immensely grateful and just happy to always have had the backup of my friends! While I am glad for the ties that continue to hold firm from my school days, I want to especially mention our Kiwiwi group, who are companions and best friends since the first days of my university education. Florian Engel, Lukas Röring, Sophie Lucas, and Melanie & Alexis Guttstadt. Thank you for the great times, for your support, and for being there (now quite literally in or around Karlsruhe). I know that I can always count on you!

My heartfelt gratitude also goes to my parents and my family. I dedicate this dissertation to my mother, Sabine, and to my father, Joachim Rietz. I am eternally thankful for your endless and unconditional love and support and for your profound belief in my abilities. Thank you for sharing this journey with me.

Finally, a very special thank you to my partner and best friend, Lena Annabell Straub, for being by my side every step of the way. Thinking back to the ups and downs of the dissertation, you deserve the biggest credit for celebrating even the tiniest ups with me, as well as guiding me through the downs. Thank you for your love, support, patience, and care. Thank you for being there for me when I needed it, for believing in me, and for a lot of excitement! I am immeasurably grateful to have you in my life.

Tim Rietz
Karlsruhe, Germany
July 2021

Abstract

With the continuously increasing impact of information systems (IS) on private and professional life, it has become crucial to integrate users in the IS development process. One of the critical reasons for failed IS projects is the inability to accurately meet user requirements, resulting from an incomplete or inaccurate collection of requirements during the requirements elicitation (RE) phase. While interviews are the most effective RE technique, they face several challenges that make them a questionable fit for the numerous, heterogeneous, and geographically distributed users of contemporary IS.

Three significant challenges limit the involvement of a large number of users in IS development processes today. Firstly, there is a lack of tool support to conduct interviews with a wide audience. While initial studies show promising results in utilizing text-based conversational agents (chatbots) as interviewer substitutes, we lack design knowledge for designing AI-based chatbots that leverage established interviewing techniques in the context of RE. By successfully applying chatbot-based interviewing, vast amounts of qualitative data can be collected. Secondly, there is a need to provide tool support enabling the analysis of large amounts of qualitative interview data. Once again, while modern technologies, such as machine learning (ML), promise remedy, concrete implementations of automated analysis for unstructured qualitative data lag behind the promise. There is a need to design interactive ML (IML) systems for supporting the coding process of qualitative data, which centers around simple interaction formats to teach the ML system, and transparent and understandable suggestions to support data analysis. Thirdly, while organizations rely on online feedback to inform requirements without explicitly conducting RE interviews (e.g., from app stores), we know little about the demographics of who is giving feedback and what motivates them to do so. Using online feedback as requirement source risks including solely the concerns and desires of vocal user groups.

With this thesis, I tackle these three challenges in two parts. In part I, I address the first and the second challenge by presenting and evaluating two innovative AI-based systems, a chatbot for requirements elicitation and an IML system to semi-automate qualitative coding. In part II, I address the third challenge by presenting results from a large-scale study on IS feedback engagement. With both parts, I contribute with prescriptive knowledge for designing AI-based qualitative data collection and analysis systems and help to establish a deeper understanding of the coverage of existing data collected from online sources. Besides providing concrete artifacts, architectures, and evaluations, I demonstrate the application of a chatbot interviewer to understand user values in smartphones and provide guidance for extending feedback coverage from underrepresented IS user groups.

Contents

List of Figures

List of Tables

List of Abbreviations

A	Attributes
ACV	Attribute-Consequences-Values
AI	Artificial Intelligence
AIM	Aggregate Implication Matrix
AL	Active Learning
C	Consequences
CASA	Computers-are-Social-Actors
CS	Computer Science
HCI	Human-Computer Interaction
HVM	Hierarchical Value Map
ICR	Intercoder Reliability
IML	Interactive Machine Learning
IS	Information Systems
ISD	Information Systems Development
ISR	Information Systems Research
LB	Laddering with Ladderbot
LD	Levenshtein Distance
MEC	Means-End Chain
ML	Machine Learning
MNB	Multinomial Naive Bayes
NLP	Natural Language Processing
PP	Paper-and-Pencil (Laddering)
QDA	Qualitative Data Analysis
QDAS	Qualitative Data Analysis Systems
RE	Requirements Elicitation
RQ	Research Question
SGD	Stochastic Gradient Descent
SiS	Similarity Scores
SRS	Software Requirements Specification
SS	Social Science
SVC	Support Vector Classifier
SVM	Support Vector Machine
TAM	Technology Acceptance Model

UCD .... User-Centered Design
UoA .... Unit-of-Analysis
V .... Values
VPP .... Visualized Paper-and-Pencil (Laddering)

1. Introduction [1]

1.1 Motivation

As the world becomes more digital every year, information systems (IS) are increasingly shaping our professional and personal lives (Villela et al., 2018). Through the Internet, IS can connect millions of geographically dispersed and culturally diverse users. Consequently, the digital transformation is influencing an ever greater part of everyone's business and private lives, changing traditional work processes and society itself (Villela et al., 2018). We are seeing a shift towards a digital society where services are developed by people for people, often using mechanisms from the internet (e.g. co-creation, crowdsourcing) to disrupt traditional businesses (Hedman et al., 2019; Kujala & Väänänen-Vainio-Mattila, 2009; Leimeister et al., 2014; Tuunanen & Peffers, 2018). In digital societies, companies must understand users and their preferences as a decisive factor for the development of innovative and successful solutions (van den Hoven, 2017). For many applications, the competitor is only a click away, which speaks for a shift of power towards the user (Leimeister et al., 2014). Successful IS are often personalized, context-adaptive, work in real-time, available anywhere, and fun to use (Leimeister et al., 2014). Organizations adopted user-centered design principles to strengthen user involvement during the development, maintenance, and evolution of IS (Brhel et al., 2015; Gasson, 2003; Maalej, Nayebi, et al., 2016; Mao et al., 2005). User-centered design (UCD) places the goals and needs of a system's end-users in the focus of the development. In UCD, it is imperative to continuously involve end-users during software development and evolution to iteratively refine prototypes and design concepts (Henfridsson & Lindgren, 2010). Moreover, other principles that guide software evolution, such as agile software development (Meth, Mueller, et al., 2015), or design thinking (Maedche, Botzenhardt, et al., 2013), also stress the importance of putting user values at the center stage for software offerings.

While the ideas of user-centered design are not new, the digital society has changed the scale at which users can and need to be involved. Frequently, IS projects fail not because of technical problems (Hofmann & Lehner, 2001), but due to inadequate catering to user needs and requirements (Ding & Liu, 2011; Neetu Kumari & Pillai, 2013), caused by lacking user involvement or incomplete information (Chakraborty et al., 2010; Tiwana & Keil, 2006). Requirements elicitation (RE) describes the act of collecting mostly qualitative data from users to understand **what** systems to build and **why** these systems matter (Tuunanen & Kuo, 2015; Tuunanen & Peffers, 2018). RE is one of the most critical and complex activities in IS development (Chakraborty et al., 2010), as many different stakeholders are involved in communicating, discussing, and negotiating requirements (Levina & Vaast, 2005).

Interviews are among the most effective techniques to involve users and other stakeholders

[1] This chapter is based on the following studies which are published: Rietz (2019), Rietz and Maedche (2019), Rietz, Toreini, et al. (2020), Rietz and Maedche (2020), Tizard, Rietz, and Blincoe (2020), Rietz and Maedche (2021a).

(Dieste & Juristo, 2011). Traditionally, practitioners conducted interviews with a well-defined small sample of users (Mohedas et al., 2015). However, as users become increasingly diverse and a single user's voice can generate invaluable insights for software evolution, an increasing number of users must be involved in development processes, with varying degrees of expertise (Jia & Capretz, 2018).

Unfortunately, interviews are costly, time-consuming, training-intensive, and location-bound (Abbasi, 2016; Deutsch et al., 2011; Meth, Brhel, et al., 2013; Miles & Rowe, 2004). These challenges make traditional manual interviews a questionable fit for the numerous, heterogeneous, and geographically distributed user groups of today (Dieste & Juristo, 2011). Furthermore, performing interviews is a complex process, prone to a lack of structure (Yamanaka et al., 2010), insufficient level of abstraction (Moitra et al., 2018), lacking interviewer confidence (Tuunanen & Rossi, 2003), and interviewer bias (Appan & Browne, 2012). A common substitute for interviews are open-ended surveys. Unfortunately, surveys are limited by participants' response behavior (Meade & Craig, 2012) and engagement (S. Kim, Lee, et al., 2019; Patton, 2002). Thus, organizations turned to explicit user feedback as a comprehensive and potentially honest source of requirements. However, the analysis of dynamic feedback sources, like social media content, struggles with data quality issues and the attributability to real users (Lappas et al., 2016; Martens & Maalej, 2019). Additionally, researchers and practitioners heavily debate dynamic data sources concerning user privacy, as organizations tend to collect and exploit data opportunistically until resistance is encountered (Günther et al., 2017). Still, organizations and users can mutually benefit from feedback being shared and combined to guide software's effective maintenance and evolution. While many software users give feedback online about the applications they use, not all users do (Tizard, Rietz, & Blincoe, 2020). Should the demographics of a user base not be fairly represented during RE, then there is a danger that the needs of less vocal users will not appropriately be considered in development. Inadequate requirements coverage risks introducing biases into systems by systematically and unfairly discriminating against certain individuals or groups in favor of others (Kujala & Väänänen-Vainio-Mattila, 2009).

Hence, building tools that enable the elicitation of requirements from a wide audience of users is crucial for developing software that meets user needs without integrating systematic biases. Thereby, such tools can contribute to reducing overall ISD project failure rates (Hofmann & Lehner, 2001; Tuunanen & Rossi, 2004). Both guidance and assistance are necessary to enable a wide audience of users to contribute requirements to development projects especially if no human interviewer is present, as users commonly are novices regarding RE processes (Mohedas et al., 2015). The requirements engineering community, in particular, has proposed several tools to tackle challenges in user involvement with diverse approaches. Predominantly, researchers focused on improving the feedback capabilities of ready-to-use software (Oriol et al., 2018; Snijders et al., 2015), simplifying involving novices with visualization-based RE (Duarte et al., 2012; Pérez & Valderas, 2009), and improving the quality of requirements (García-López et al., 2020; Li et al., 2005; Lucassen et al., 2016). Further, the various limitations with managing and performing interviews motivated the exploration of tool-support for interviewers (Bano, Zowghi, & da Rimini, 2018; Debnath & Spoletini, 2020; Elrakaiby et al., 2017; Jean-Charles & Spoletini, 2019), such as utilizing a ”stable” automated interviewer (Nunamaker et al., 2011), e.g., a chatbot. However, only a few studies looked into automated interviewers as means for elicitation, e.g., in the form of an embodied conversational agent to facilitate a group workshop aimed at user story formulation (Derrick et al., 2013).

Chatbots, text-based conversational agents powered by artificial intelligence (AI), have seen rising interest over the last years. Chatbots have the potential to assist with user interviewing and requirements elicitation (Gnewuch, Morana, & Maedche, 2017; Tallyn et al., 2018), as they can be used in various contexts, scale very well, and allow to precisely control the interview structure (S. Kim, Lee, et al., 2019). Details of interviews, such as the formulation, ordering, and omission of questions, are crucial, as is the reasoning behavior of analysts (Bano, Zowghi, & da Rimini, 2018). Analysts commonly reason based on models, while novices think in relationships between objects and attributes (I.-L. Huang & Burns, 2000). Hence, with the proper interviewing technique, a chatbot may be capable of navigating the downfalls of (human) interviewers. Chatbots have multiple benefits, some of which are their availability, learning curve, and platform independence (Klopfenstein et al., 2017). These benefits make them a good fit for involving wide audiences of users. Their availability and platform independence make for a barrier-free experience, as users can access them via their internet browser. Furthermore, chatbots provide a gentle learning curve, as users are mostly already familiar with the mode of interaction, texting.

While chatbots are the subject of many studies (Maedche, Legner, et al., 2019), their application for elicitation, either of information in general or requirements in particular, remains sparse. Previous work has largely hinted at the applicability of chatbots as interviewers to guiding workshops (Derrick et al., 2013), detect human physiology and behavior during interactions (Nunamaker et al., 2011), conduct scripted accounting interviews (Pickard, Schuetzler, et al., 2017), gathering ethnographies (Tallyn et al., 2018), market research (Xiao et al., 2020), and substituting for survey-based forms of Likert-style questions (S. Kim, Lee, et al., 2019). Overall, these studies demonstrate the potential of the utilization of chatbots for gathering information. While these studies provide valuable insights into how users react to these interviews and call for flexibility in interview structure, they provide an incomplete account of how to design semi-structured dialogue strategies. Recently, some scholars have applied chatbots specifically for the case of RE: ReqBot is a sequential and static chatbot that asks users to describe requirements for specific software. While the bot allows users to suggest ideas and requirements in a survey-like form, its focus lies on detecting ambiguities between requirements and asking for clarification (Valkenier, 2020). On the other hand, CORDULA is an early-stage proposal for a chatbot focused on interacting with users to partially compensate deficits in user requirements (Friesen et al., 2018). On a grand scale, however, current approaches to chatbots for RE evolve around using a survey-like approach to asking questions while focusing on implementing approaches to improve the quality of collected requirements. Thus, elicitation chatbots are far from providing an experience similar to a human-conducted interview. For utilizing interview chatbots, it is imperative to identify appropriate interview techniques that lend themselves to automation and compare the approach against survey-based methods for user involvement (Dieste & Juristo, 2011).

So far, I outlined how chatbots are a promising approach to involve a wide audience of users in Information Systems Development (ISD). However, the prospect of creating large datasets containing numerous interviews leads to a subsequent challenge: making sense of a large amount of unstructured text. This challenge is especially severe in qualitative studies, e.g., as part of a broader RE process. Here, analysts perform qualitative coding by annotating text with short labels to make sense of the data. Qualitative coding is highly valuable to produce a nuanced understanding of a dataset to answer explorative or investigative questions based on the underlying qualitative data (Ritchie & Lewis, 2003). Specifically, analysts are trying to answer *why?*- and *how?*-questions when working with qualitative data. Qualitative coding has been described as both art and science, and as such, requires intensive training and experience from analysts (Ritchie & Lewis, 2003). While qualitative coding is time-consuming, even for small datasets, the process becomes unreliable and intractable with large amounts of data (Abbasi, 2016; N.-C. Chen, Drouhard, et al., 2018). Manual coding is severely limited by the available workforce (Crowston, Allen, et al., 2012). For example, Xiao et al. (2020) used a chatbot asking open-ended questions to collect over 11.000 free-text responses, of which only 50% could be analyzed through qualitative coding in a reasonable time frame. Additionally, much of the coding process can become repetitive and painstaking, particularly after creating an initial codebook during the first iteration of the iterative coding process (Marathe & Toyama, 2018).

While automating the entire analysis process might seem appealing, Marathe and Toyama (2018) report from an interview study that researchers performing qualitative data analysis desire support from a system only after developing an initial codebook based on parts of the dataset. Some degree of automation is already integrated into the big players in coding software (NVivo, Atlas.ti, MAXQDA), such as suggesting labels based on the available labeled examples created during coding. However, the integration of these features into the coding process lacks transparency and customizability (N.-C. Chen, Drouhard, et al., 2018), resulting in a lack of trust in automated suggestions (N.-C. Chen, Kocielnik, et al., 2016; Drouhard et al., 2017), difficulties in mastering the complex analytical features (Marathe & Toyama, 2018), and overall little support for speeding up the coding process or improving coding quality (Marathe & Toyama, 2018; Sánchez-Gómez et al., 2019).

Several success stories showcase how machine learning (ML) techniques can support certain aspects of qualitative data analysis. For example, ML can help with identifying potentially ambiguous data during coding (Drouhard et al., 2017), identify document sections for a specific label with a high recall by using expert-defined rules for coding (Crowston, Allen, et al., 2012; Grimmer & Stewart, 2013), and provide reliable code suggestions with enough training data (Yan et al., 2014). The machine-teaching paradigm of interactive machine learning (IML) seems particularly promising for increasing coding productivity (N.-C. Chen, Drouhard, et al., 2018). In IML, a user iteratively builds and refines an ML model in a cycle of teaching and refinement (Dudley & Kristensson, 2018). The iterative training is similar to the iterative refinement of codes and coding rules during qualitative coding and can help analysts build trust in ML-based recommendations (Marathe & Toyama, 2018).

Despite the potential of utilizing (interactive) machine learning to support analysts (and other users of analysis tools for qualitative data, e.g., qualitative researchers) with coding and understanding large datasets, recent users studies demonstrated two significant shortcomings that restrict the value of available approaches (N.-C. Chen, Kocielnik, et al., 2016; Drouhard et al., 2017; Marathe & Toyama, 2018). Firstly, the integration of ML into the analysis needs to be improved by enabling users to refine code suggestions iteratively. Often, systems limit the interaction to accepting and rejecting suggestions, rather than nourishing an interaction where users and systems support each other in improving coding and suggestion quality (N.-C. Chen, Drouhard, et al., 2018). Secondly, systems need to increase the transparency of suggestions to enable researchers to understand and reproduce a system's behavior and report the coding process in sufficient detail for a scientific publication (Grimmer & Stewart, 2013; Marathe & Toyama, 2018).

To summarize, with the continuously increasing impact of IS on private and professional life, it is crucial to integrate users into IS development processes in a scalable way. However, existing requirements elicitation and analysis techniques and tools come with several limitations. I observe three significant challenges: **First**, existing elicitation techniques such as interviews do not scale for wide audience user groups. **Second**, even if qualitative data can be collected from a wide audience, the resulting datasets' size limits the applicability of established methods and tools for qualitative data analysis. In this thesis, I present two innovative solutions two tackle the issues raised by the data collection and analysis from wide audiences. Specifically, I present a chatbot for requirements elicitation and an AI-based system to semi-automate coding. **Third**, while organizations elicit requirements from user feedback in online channels, little is known about the demographics of who is giving feedback and what motivates them to do so. Thus, I present an in-depth demographic study of software feedback engagement. I investigate the demographics, feedback habits, and users' willingness to utilize new ways to be involved in IS development. The following section translates the challenges and strategies for wide audience involvement into overarching Research Questions (RQs).

1.2 Research Gaps and Research Questions

This thesis explores AI-based system support for qualitative data collection and analysis and supports researchers with collecting and understanding unstructured, natural language data from wide audiences. Therefore, I define four RQs that I addressed with four studies presented in this thesis, as shown in Figure 1.1. I present these research questions in the following.

The **first RQ** deals with the design of a system for the scalable collection of qualitative data through requirements elicitation interviews.

As users commonly have little experience with contributing requirements, it is necessary to understand how to support novice users during elicitation interviews. However, RE literature

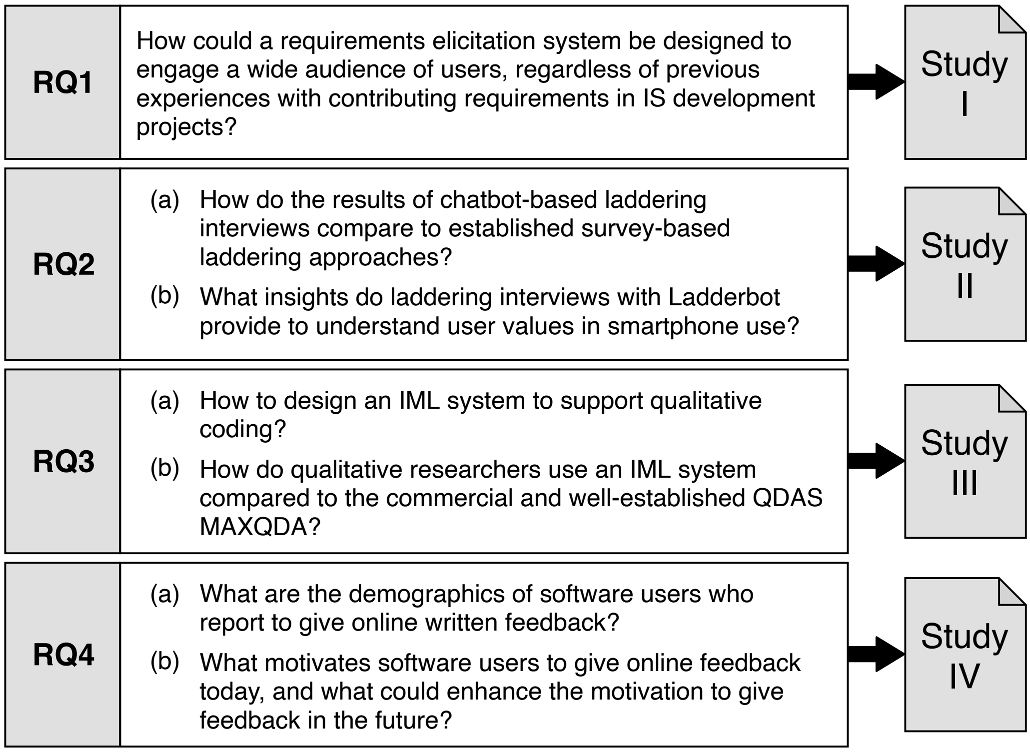


Figure 1.1: Overview of research questions addressed in this thesis.

rarely focuses on supporting novice users (Villela et al., 2018), as novice RE analysts are the focus of supporting activities (Bano, Zowghi, Ferrari, et al., 2018; Jean-Charles & Spoletini, 2019). Research needs to identify appropriate elicitation techniques that can provide the structure and level of abstraction required to include novices (Moitra et al., 2018; Yamanaka et al., 2010). For eliciting relevant information, the laddering interview is a very effective technique (Dieste & Juristo, 2011). Laddering produces comprehensive and structured insights and allows the interviewer to understand the hierarchical goal structure that links concrete means to abstract ends. During laddering interviews, interviewers start with an initial topic and as a series of *why?*-questions to better understand user experiences by uncovering linkages of needs and values (Deutsch et al., 2011; Vanden Abeele et al., 2012). However, laddering interviews require well-trained interviewers to assist users that are struggling to find an answer (Deutsch et al., 2011). Additionally, the technique fosters boredom and fatigue due to its repetitive question structure (Kaciak & Cullen, 2009). While chatbots may be able to address these shortcomings, the integration of chatbots into RE remains sparse, especially as automated interviewers for RE. To close this gap, I address the following RQ in **Study I**:

RQ1: *How could a requirements elicitation system be designed to engage a wide audience of users, regardless of previous experiences with contributing requirements in IS development projects?*

I answered this RQ by aggregating common issues in user elicitation interviews, mapping the benefits and difficulties of laddering interviews to the identified issues, and proposing a design and an architecture for a chatbot capable of conducting laddering interviews.

The **second RQ** deals with evaluating a laddering interview chatbot by assessing the quality of elicited information for a dedicated (research) use case. Further, it compares the results to established methods for wide audience laddering interviews based on descriptive, quantitative, perception-based, and content-based measures.

Laddering interviews feature multiple strengths, such as providing a structure for quantifying and analyzing qualitative data (Rugg et al., 2002b), allowing a detailed analysis of usage motives and cognitive structures of users (Wilhelms et al., 2017), and explaining relationships between goals (Jung, 2014). Despite its strengths, laddering interviews have gained little traction in IS journals and conferences (Rzepka, 2019; Tuunanen & Kuo, 2015). Predominantly, studies applying the laddering interview technique present the same shortcomings: limited sample sizes, which create homogeneity or sparsity in ages and demographics in participant samples (Gao et al., 2019; C. F. Lin et al., 2020; Rzepka, 2019). As laddering interviews scale poorly to wide audiences, researchers rely on survey-based laddering methods when aiming to achieve a large sample size (Jung, 2014; Miles & Rowe, 2004). However, this method faces multiple limitations: It restricts interviewees' responses (Pieters, Bottschen, et al., 1998; Russell, Flight, et al., 2004), provides little assistance in the case of misunderstandings or problems (Miles & Rowe, 2004), and fosters boredom and fatigue due to a repetitive question structure (Kaciak & Cullen, 2009). As a laddering chatbot may overcome these shortcomings, I address the following RQ in **Study II**:

RQ2a: *How do the results of chatbot-based laddering interviews compare to established survey-based laddering approaches?*

While comparing chatbot-based and survey-based laddering based on quantitative, interaction-, and perception-based measures can help evaluate the applicability of chatbot interviewers, some studies have already shown promising results of supplementing surveys with chatbots (S. Kim, Lee, et al., 2019; Nunamaker et al., 2011; Tallyn et al., 2018). However, these studies have not investigated the data quality from chatbot interviews regarding its value for a research or industry project. Aiming to close this gap, I utilize the laddering interview chatbot design from RQ1 (Ladderbot) to understand how user values in smartphone use changed. Therefore, I compare the results of wide audience laddering interviews with findings from manual laddering conducted in 2014 (Jung, 2014) to answer the following second RQ with **Study II**:

RQ2b: *What insights do laddering interviews with Ladderbot provide to understand user values in smartphone use?*

I address RQ2a and RQ2b by conducting laddering interviews with 256 smartphone users using two survey-based and one chatbot-based laddering approach (with the Ladderbot system). I analyze the data to understand users' hierarchical value structure and participants' perception and behavior of the individual data collection approaches. Further, I compare the three approaches to laddering based on quantitative and qualitative results, report insights on positive and negative impacts of smartphones, and discuss the strengths and weaknesses of online laddering surveys and chatbots for wide audience involvement.

The **third RQ** deals with evaluating an AI-based system that semi-automates the coding step of qualitative data analysis with qualitative researchers.

Existing systems to support QDA provide only limited automation capabilities for coding. For example, systems such as Nvivo or INCEpTION make code recommendations using ML. Simple approaches to making recommendations use keyword- or structure-matching to highlight sections based on user- or system-generated keywords. More sophisticated approaches use user-generated annotations to train an ML model through supervised learning (Klie et al., 2018). However, user-centered studies suggest that ML-based automation capabilities do not meet user expectations (Marathe & Toyama, 2018). Primarily, existing implementations fail to provide explanations for recommendations, thus lacking transparency (N.-C. Chen, Drouhard, et al., 2018). As a consequence, researchers lack trust in automated coding (Drouhard et al., 2017). Furthermore, functionality for revising recommendations is mainly limited to accepting or rejecting a code and does not help researchers with identifying flaws in codebooks or in the code rules they follow. With the lack of transparent recommendations and limited capabilities for iteratively revising code rules to train an ML-based system, qualitative researchers are reluctant to adopt ML-based support for qualitative coding (Marathe & Toyama, 2018). To close this gap, we need to understand better how researchers interact with AI-based coding support systems and compare the interaction with available and established QDAS. Therefore, I answer the following RQs in **Study III**:

RQ3a: *How could an IML system be designed to support qualitative coding?*

RQ3b: *How do qualitative researchers use an IML system compared to the commercial and well-established QDAS MAXQDA?*

I address RQ3a and RQ3b by designing and developing an AI-based system to semi-automate qualitative coding for qualitative research. Therefore, I aggregate relevant literature on qualitative coding and AI-based coding support to develop six design requirements. I instantiate these requirements in a system prototype that integrates rule-based coding and supervised ML, which I evaluate in two studies with 17 qualitative researchers. I compare the researchers' interaction with the prototype against the interaction with MAXQDA and present insights into how researchers work with automated suggestions. Additionally, I analyze how researchers feel about transparency features for suggestions and how suggestions impact their coding agency.

The **forth RQ** deals with dynamic feedback sources for RE, by investigating users' demographics and feedback engagement on the three most prominent online channels: app stores, product forums, and social media.

Organizations and development teams rely on such online feedback to elicit requirements from what has been called the *voice of the users* (Guzman, Alkadhi, et al., 2016; Guzman, Alkadhi, et al., 2017; Pagano & Maalej, 2013; Tizard, Wang, et al., 2019). Recent literature heavily studied efficient methods to extract requirements insights from this "voice" (Guzman, Ibrahim, et al., 2017; Sorbo et al., 2017), yet very little research has investigated the demographics of who is giving feedback in these channels. Demographic data of the users giving feedback is usually not included in these channels to support the privacy of feedback givers. As such, little is known about the diversity of the voice of the users, bearing the risk of including the concerns and desires of vocal users and user groups in development decisions solely. Consequently, there is a gap in understanding which users give online feedback and which groups are underrepresented today to develop the best solutions for user integration into IS development processes tomorrow. To fill this gap, I address the following RQ in **Study IV**:

RQ4a: *What are the demographics of software users who report to give online written feedback?*

Additionally, previous work identified discrepancies between the feedback behavior that users expected of themselves and their actual feedback rate in the real world (Stade et al., 2020). The study also suggested that smart assistant facilitation of feedback elicitation may encourage feedback compared to traditional methods. Thus, it is essential to understand why software users decide to give online feedback and how new methods would potentially impact feedback behavior. As I investigated the effects of a chatbot interviewer for encouraging more answers in Study II, Study IV investigates the perception of new data collection methods and compares users' motivations to give feedback across demographics and usage behavior. Thus, I address the following second RQ in **Study IV**:

RQ4b: *What motivates software users to give online feedback today, and what could enhance the motivation to give feedback in the future?*

I answer RQ4a & b by conducting two surveys of software users from Germany, New Zealand, and China, including 1976 complete responses. Based on the collected responses, I present insights on which software users give online feedback, what motivates users when they give feedback, and discourages them when they do not.

1.3 Thesis Structure

Figure 1.2 shows the outline of this thesis consisting of six chapters. **Chapter 1** motivates the topics and introduces the relevant research gaps as well as the central research questions that the thesis addresses. **Chapter 2** presents the foundations relevant for this thesis, including the role of qualitative data in IS development and research, qualitative data collection, qualitative data analysis, and AI-based technology for qualitative data collection and analysis.

Chapter 3 includes **part I** of this thesis and focuses on AI-based systems for qualitative data collection and analysis in ISD. Part I includes three studies. In *Study I*, I propose a design and an architecture for a chatbot for requirements elicitation interviews using the laddering interview technique (RQ1). In *Study II*, I present the evaluation of the chatbot design outlined in Study I by performing online chatbot- and survey-based laddering interviews with 256 participants in three treatments on user values in smartphone use. The findings from Study II highlight the strengths and weaknesses of chatbot-based laddering and outline strategies for wide audience laddering interviews (RQ2a). Further, Study II presents a hierarchical map of goals and values of smartphone use (RQ2b). Inspired by the large dataset collected in Study II, *Study III* presents an IML system to semi-automate qualitative coding. While conceived initially as tool-support for laddering interview analysis, I expanded the IML-system to support qualitative coding of all kinds of qualitative data. Study III outlines both design requirements for IML-based coding systems and introduces Cody as a prototype system instantiating the outlined requirements (RQ3a). Further, in Study III, I present the results of a formative (n=6) and a summative (n=11) evaluation of the prototype with qualitative researchers, which compares the prototype to the established QDAS MAXQDA (RQ3b).

Chapter 4 includes **part II** of this thesis and focuses on software users as a source of feedback and requirements in IS development and includes one study. While part I investigates using novel and innovative artifacts for data collection and analysis, part II shifts the attention from artifacts to the user. Rather than developing artifacts top-down, I approach users bottom-up to explore demographics, as well as motivations for contributing feedback. In *Study IV*, I present results from two large-scale survey studies on user feedback engagement with 1040 (survey I) and 936 (survey II) software users from Germany, China, and New Zealand. Thereby, Study IV sheds light on who gives feedback in app stores, software forums, or on social media (RQ4a), and presents findings on what motivates and discourages user feedback, as well as strategies for encouraging feedback (RQ4b).

Chapter 5 summarizes the results of this thesis by highlighting and discussing theoretical contributions and practical implications. Furthermore, I present the limitations of this thesis and provide avenues for future work. **Chapter 6** concludes this thesis.

Parts of this thesis have been published in IS, HCI, or RE outlets. In addition, some sections of this thesis are in preparation for submission or under review. I indicated the corresponding publications at the beginning of each chapter. A list of publications, papers under review, and working papers in preparation for submission are listed starting on page 164.

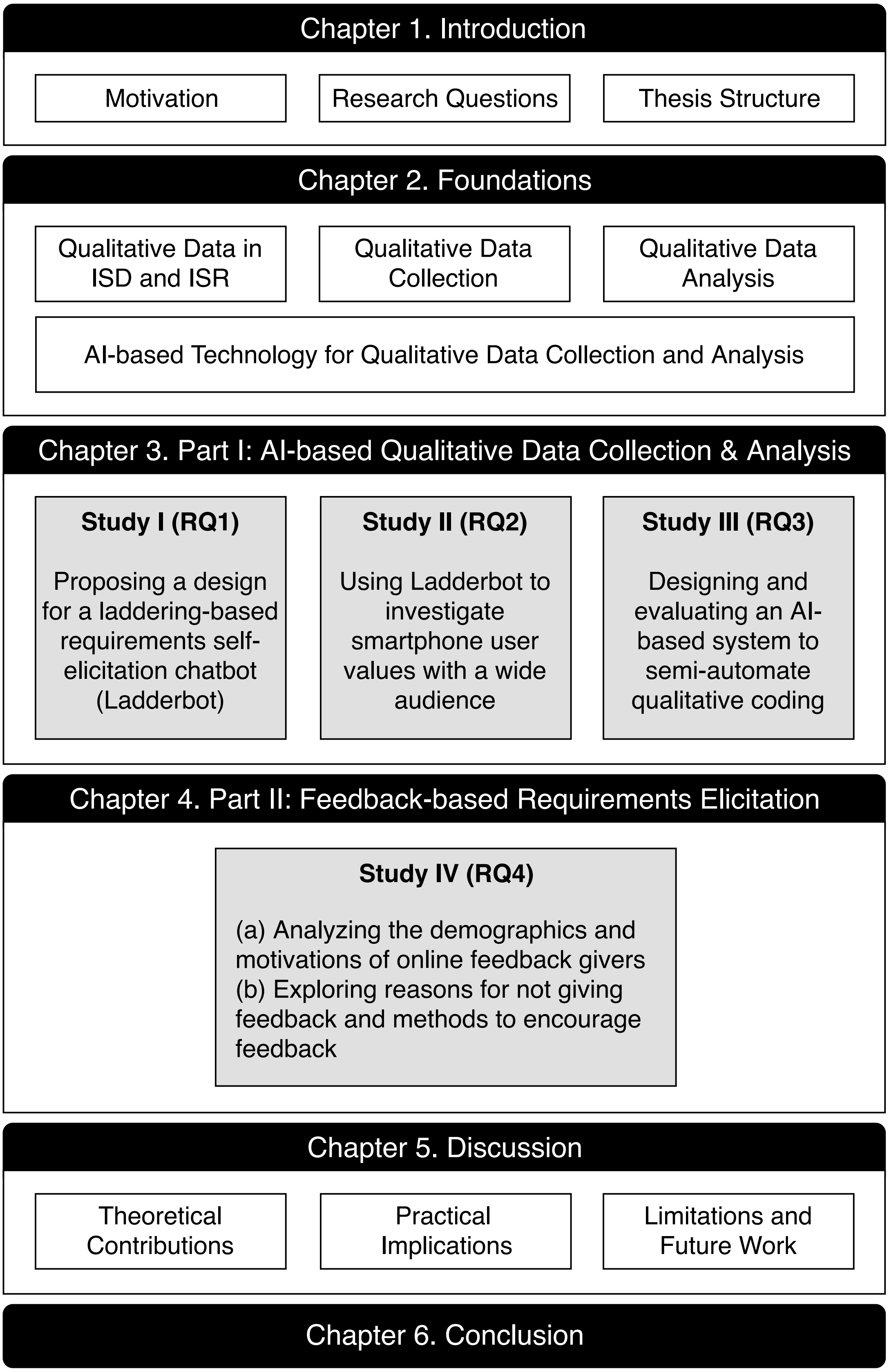


Figure 1.2: Structure of the thesis.

2. Foundations [2]

In this thesis, I focus on the intersection between three larger research streams, including qualitative data collection, qualitative data analysis, and AI. Figure 2.1 presents an overview of selected research streams with selected example studies. Additionally, the research gaps introduced in Section 1.2 are positioned within these research streams.

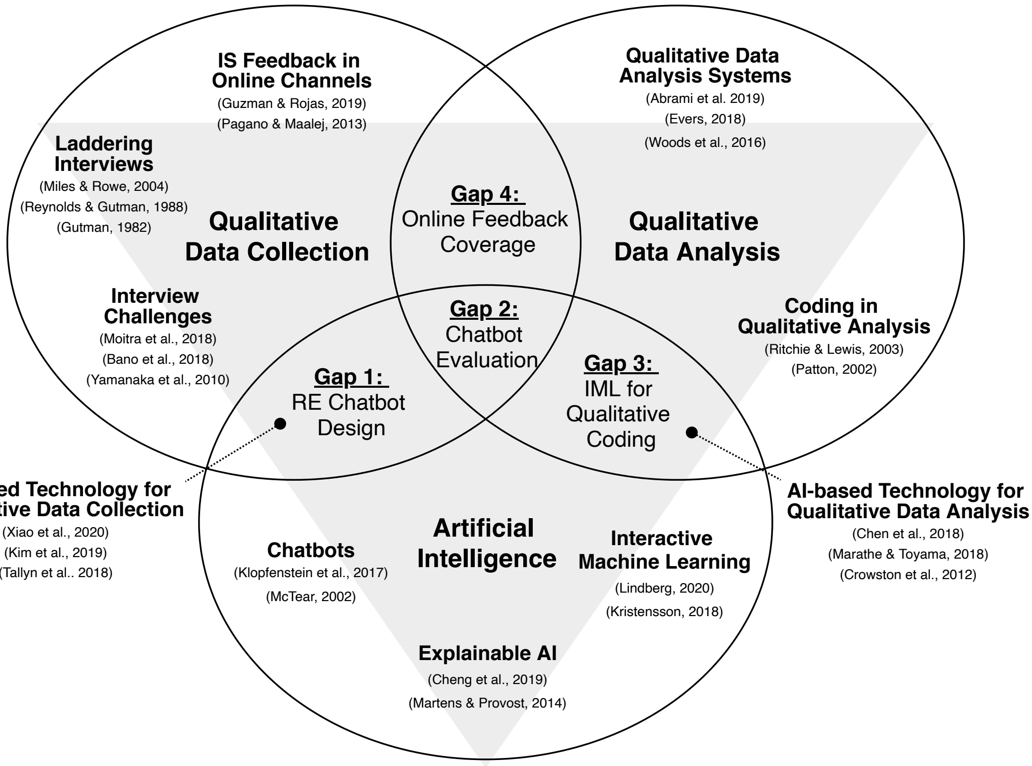


Figure 2.1: Overview of foundations and research gaps.

I approach the stream of qualitative data collection primarily from an IS and RE perspective. In particular, I investigate laddering interviews, which are used to elicit hierarchical relationships between concepts from participants (Miles & Rowe, 2004). While laddering interviews have multiple benefits, they also face several interview challenges, restricting their applicability with wide audiences (Moitra et al., 2018). The first research gap is positioned at the intersection of qualitative data collection and the second stream, AI. In the AI stream, I investigate AI-based technology for qualitative data collection, especially chatbots. The second research gap addresses the intersection of the first two with the third stream, qualitative data analysis, by evaluating a chatbot as means for wide audience interviewing. I rely on foundations from qualitative data analysis systems and coding in qualitative analysis to understand promises and shortcomings of the coding process and available tool-support. The third research gap lies at the intersection of AI and qualitative data analysis. From the AI stream, I utilize the sub-streams of IML, explainable AI, and

[2]This chapter is based on the following studies which are published or in work: Rietz and Maedche (2019), Rietz and Maedche (2020), Rietz and Maedche (2021a), Rietz and Maedche (2021b), Tizard, Rietz, Liu, et al. (2021).

AI-based technology for qualitative data analysis to investigate gap three. Finally, research gap four addresses the intersection between qualitative data collection and qualitative data analysis by analysing the feedback coverage of software users and their motivations to provide feedback.

2.1 The Role of Qualitative Data in IS Development and Research

Qualitative inquiry cultivates the most useful of all human capacities: The capacity to learn
– Halcolm's Law of Inquiry in (Patton, 2002, p. 1)

One can distinguish three kinds of qualitative data based on the respective method of data collection, following Patton (2002): (1) in-depth, open-ended interviews, (2) direct observation, and (3) written documents. *Interviews* are usually recorded and transcribed, and produce direct quotations from people about their experiences, feelings, opinions, and knowledge, including rich context for interpretation. Data from *observations* comes in the form of field notes and contains detailed descriptions of people's activities, behaviors, and actions. Further, observations can contain descriptions of interpersonal relations and organizational processes, depending on the goals of the qualitative inquiry. *Documents* contain written material and multimedia documents from organizational, program, or private records. This includes, amongst others, official publications, personal diaries, letters, photographs, video recordings, or written responses to open-ended surveys. Qualitative findings can be presented alone or combined with quantitative data. At the simplest level, an interview or a questionnaire asks both closed and open-ended questions, thus combining quantitative measurement and qualitative inquiry (c.f. Patton (2002)). In a research context, qualitative methods are used to fulfil one or multiple of the following activities (from Ritchie and Lewis (2003)):

- Contextualize - describing the form or nature of what exists
- Explain - examining the reasons for, or associations between, what exists
- Evaluate - appraising the effectiveness of what exists
- Generate - aiding the development of theories, strategies or actions.

Skillful interviewing involves much more than just asking questions. Content analysis requires considerably more than just reading to see what's there. Generating useful and credible qualitative findings through observation, interviewing, and content analysis requires discipline, knowledge, training, practice, creativity, and hard work.
– From Patton's Qualitative Research & Evaluation Methods (Patton, 2002, p. 5)

Qualitative research usually takes the following steps: (1) defining a research question, creating a research design by defining a setting, selecting a time frame, and choosing a data collection method, (2) designing and selecting participant samples, (3) designing a fieldwork strategy and materials, (4) collecting qualitative data, (5) carrying out qualitative analysis, (6) generalizing from qualitative research, and (7) reporting and presenting qualitative data (for in-depth guidelines and discussions of the individual steps, see Patton (2002) and Ritchie and Lewis (2003)). Looking at IS research, interviews in particular used to be a largely unexamined data collection technique (Myers & Newman, 1999). The application of structured interviewing methods, in particular, was lacking in ISR, with room for improvement in designing, conducting and reporting interview-based research (Schultze & Avital, 2011). Recently, the number of qualitative studies in ISR has been growing (Sarker et al., 2013; Stafford & Farshadkah, 2020).

ISD is the IS field's oldest subarea (Klein, 2003) and conceived as the defining core of the field with historically as much as half of all research relating to ISD (Hassan & Mathiassen, 2017; Morrison & George, 1995). Arguably at the center of the ISD environment and one of the key reasons for failed ISD projects is the RE step (Chakraborty et al., 2010). RE in ISD commonly involves communication and knowledge transfer between an analyst and a user, in which the analyst (attempts) to build an understanding of the user's needs (Browne & Rogich, 2001). Therefore, analysts may structure underlying problems into (organizational) goals, (business) processes, tasks that have to be performed to achieve the goals, and information (data) that is necessary to inform task behaviors (Yadav et al., 1988).

Numerous techniques can be used for RE, each with individual strengths and weaknesses (see Table 2.1 for common techniques). Arguably the most commonly use technique is interviews (Bano, Zowghi, Ferrari, et al., 2018; Pickard & Roster, 2020). The requirements documented in a requirements document usually stem from an exchange between an analyst and a user, typically through interviews and workshops (Maalej, Nayebi, et al., 2016). Recently, organizations also started utilizing *user feedback* as requirements source, which is for example collected through social media channels, user forums, review-, or crowd-feedback systems (Maalej, Nayebi, et al., 2016; Xu et al., 2015). Feedback can be distinguished into *explicit user feedback*, provided by users after interacting with the software in visual or readable expressions (e.g., text and emoticons), or *implicit user feedback*, in a nonverbal format obtainable through monitoring application usage and context (C. Wang, Daneva, et al., 2019). Table 2.1 provides an overview of commonly used requirements elicitation techniques and feedback sources for requirements elicitation. When an analyst believes to have build a sufficient understanding of the user's needs, the information is recorded in a requirements document. The requirements document contains information elicited from users and other sources and represents a description of a system that is aimed at enabling the user to achieve the goals identified (Browne & Rogich, 2001). There exist various approaches to documenting or specifying requirements, such as prototyping, sequence diagrams, feature models, or user stories (see Jarzebowicz and Polocka (2017)). Users stories are one of the most timely representations of requirements, due to their integral role in several Agile methodologies, including XP and Scrum (Beck & Fowler, 2000; Cohn, 2004). A user story is a short, one or two sentence account in the user's own words of how the user would like to use the software, following the form:

As a <type of user>, I want <some goal> so that <some reason>

Domain		Technique	Explanation
Techniques and Approaches for Requirements Elicitation (Yousuf & M.Asger, 2015; Zowghi & Coulin, 2005)		*Interviews*	Most traditional and commonly used technique for RE, direct conversation between analyst and user, can be distinguished into unstructured, semi-structured and structured interviews
		Questionnaires	Mainly used in early stages of RE, consist of open and/or closed questions
		Task Analysis	Used to construct a hierarchy of the tasks performed by the users and the system, and determine the knowledge used or required to carry them out
		Domain Analysis	Investigation of the existing and related documentation and applications. Performed to extract early requirements and understand and capture domain knowledge
		Introspection	Analyst develops requirements based on their own perception and believes about what users and other stakeholders want and need
		Repertory Grids	Ask users to develop attributes of a system and assign values these entities to identify similarities and differences between domain entities
		Card Sorting	User sorts a series of cards containing the names of domain entities into groups according to their own understanding
		Laddering	Users are asked a series of short prompting questions known as probes, to arrange knowledge in a hierarchical fashion
		Group Work	Collaborative meetings with analysts and multiple users to involve and commit stakeholders directly and promote cooperation
		Brainstorming	Participants rapidly generate as many ideas as possible in an informal discussion
		Joint Application Development	Involve all available stakeholders into a discussion of the problems to be solved and the available solutions to those problems
		Workshops	Generic term for a number of types of group meetings where multiple stakeholders cooperate on developing and discovering requirements
		Ethnography	Study of people in their natural setting where the analyst actively or passively participates in normal activities of the user over an extended period of time
		Observation	One of the more widely used ethnography techniques, analyst observes the actual execution of existing processes without interference
		Protocol Analysis	Participants perform a task whilst talking it through aloud, describing the conducted actions and the thought process behind them
		Apprenticing	Analyst learns and performs the tasks under the instruction and supervision of an experienced user
		Prototyping	Providing stakeholders with prototypes of a system to support the investigation of possible solutions
		Goal Based Approaches	Objectives of a system are decomposed into AND/OR relationships and elaborated with why and how questioning
		Scenarios	Narrative and specific descriptions of current and future processes including actions and interactions between the users and the system
		Viewpoints	Model the domain from different perspectives to develop a complete and consistent description of the target system
Feedback Sources for Requirements Elicitation (Lim et al., 2021)	Explicit user feedback	*Online Reviews*	Include app reviews, reviews compiled by experts, and online user reviews, commonly short texts that describe a usage experience with no particular structure
		Microblogs	Data from Twitter, Facebook, and Weibo, including metadata such as likes, number of retweets, and hashtags, usually in a very short textual form
		Online Discussion/Forum	Online forum posts from dedicated websites such as feature tracker, open-source forums, and online forums such as Reddit
		Software Repositories	Feedback in online software repositories include, e.g., Apache OpenOffice, GitHub, JIRA, Jenkins
		Software/App Product Descriptions	Software product descriptions on app description pages or app change logs
	Implicit user feedback	*Sensor Readings*	Data collected during the usage of a software, e.g., location and motion state of a mobile device, camera data
		Usage Data	Usage data collected by the software during the interaction with feature functions, e.g., click paths, visited sites, used functions
		Mailing Lists	Open-source lists of users of software, e.g., the Apache Common User List

Table 2.1: Commonly used techniques for RE and sources of feedback.

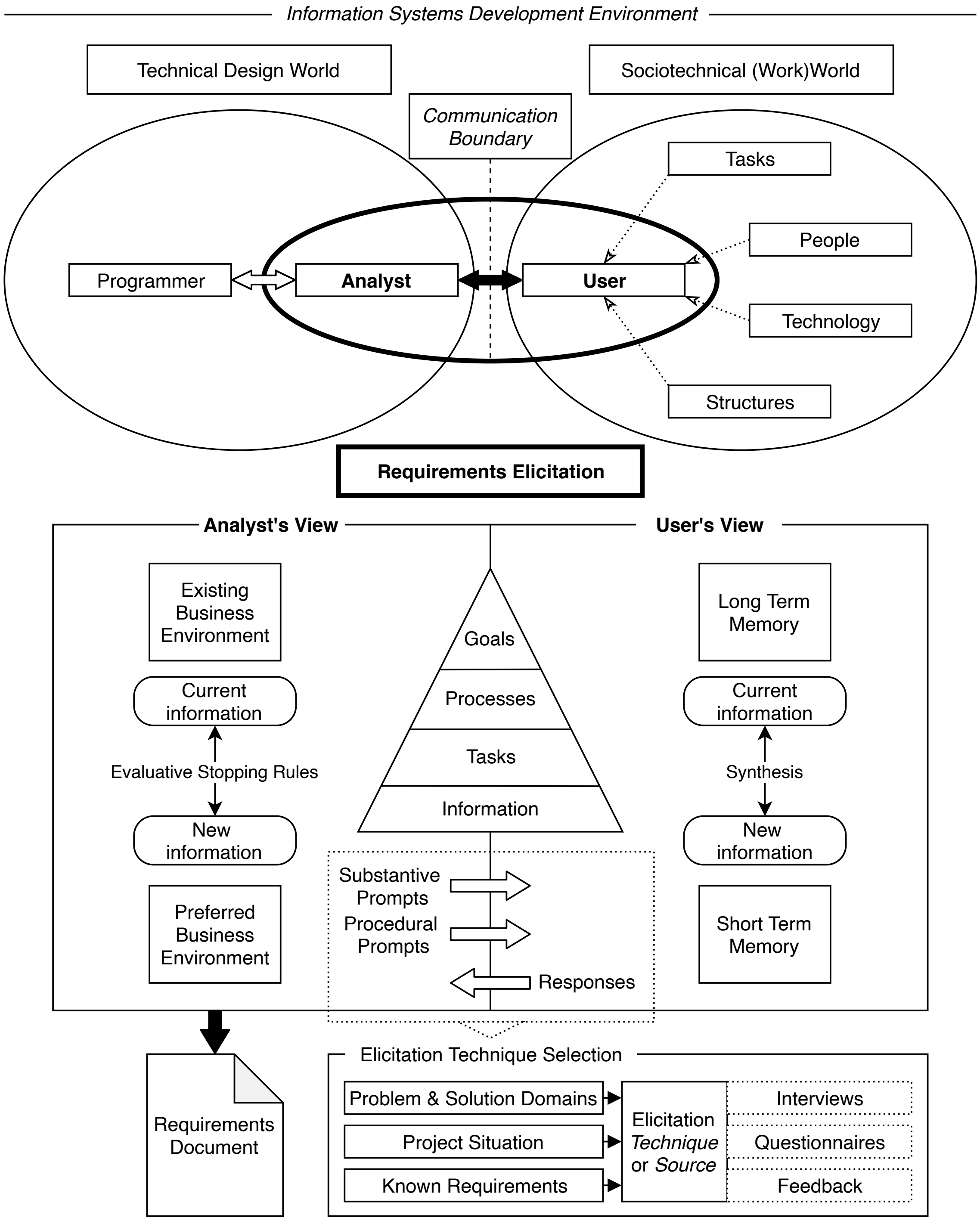


Figure 2.2: The information systems development environment and its relation to the requirements elicitation task model, including details of elicitation technique selection. Own visualization adapted from Browne and Rogich (2001), Chakraborty et al. (2010), Garrity (2001), and Hickey and Davis (2004)

Figure 2.2 shows the ISD environment, its relation to RE, the RE task model, and the influencing factors for elicitation technique selection (Garrity, 2001). To summarize, RE is positioned at the intersection between analysts and users (Chakraborty et al., 2010). The analyst is concerned with developing a vision for the technical artifact by starting with the available information on the existing business environment (current state) and using new information to develop a vision of a preferred business environment (future state) (Browne & Rogich, 2001). Since the analyst will not be able to utilize new information in its entirety, the analyst will use evaluative stopping rules to distinguish relevant from discarded information (Goel & Pirolli, 1989). The user, on the other side of the communication boundary, is influenced by tasks, people, technology, and structures. During the RE process, the user iterates in a synthesis process where information from the user's long term memory (current information) is enhanced and shaped by new, short term memory experiences as part of the RE interaction (new information) (Browne & Rogich, 2001). RE is shaped by an reciprocal interaction consisting of analyst prompts and user responses. Eventually, the form of both prompts and responses is determined by the underlying elicitation technique. Problem and solution domains, the project situation, as well as the amount and details of known requirements shape the technique selection (Hickey & Davis, 2004).

While Figure 2.2 provides a broad overview of the ISD environment as well as an orientation towards the role and the process RE in ISD, this thesis focuses on sub-sections of this environment. Particularly, I am interested in three aspects: Firstly, I focus on the arrow connecting the analyst and the user. While the figure implies a 1-1 relationship between the actors, I investigate ways to extend this relationship to 1(analyst)-n(users). Second, I am concerned with assisting the analyst in obtaining new information through the analysis of large qualitative data sets. Thirdly, I analyse feedback as an elicitation source regarding its coverage of individual user groups, as well as users' motivations to provide feedback.

2.2 Qualitative Data Collection

There are numerous techniques for researchers and practitioners to choose from to collect qualitative data. The most commonly use techniques in both domains, arguably, are interviews and open-ended surveys (Bano, Zowghi, Ferrari, et al., 2018; Patton, 2002; Pickard & Roster, 2020; Ritchie & Lewis, 2003). In the following, I introduce one interview technique in detail, laddering interviews, and outline common interview challenges, particularly with novice interviewees. Furthermore, I explain benefits and shortcomings of using feedback in online channels as a requirements source.

2.2.1 Laddering Interview Technique

Laddering initially stems from personality psychology to utilize a structured approach to data-gathering (Miles & Rowe, 2004). It was introduced as a method to elicit superordinate items from subordinate ones, to clarify the relations between items obtained using the repertory grid method[3], with its origin in personal construct theory. However, the laddering technique has primarily been used for knowledge-elicitation in marketing and advertising, know as the *means-end chain* (MEC) approach (Tuunanen & Rossi, 2004). The MEC

[3]The *repertory grid method* is a technique to elicit personal constructs, such as good-evil, happy-sad, that determine how a person sees the world. In the method, a participant is presented with groups-of-three of, e.g., important figures in their life. The participant is asked to say in what way two are alike, but different from the third. This process is repeated until the participant has produced all the constructs or the investigator stops the process. While the repertory grid procedure identifies constructs, it does not provide information about the hierarchical relationship between these.

approach is also frequently applied in research oriented on user values, due to it enabling a systematic structuring of results by providing an approach to quantifying results (Wilhelms et al., 2017). The approach assumes that the attributes of products or services are means for customers to achieve values, which become subsequent means to achieving another higher goal or value (Gutman, 1982; Miles & Rowe, 2004; Reynolds & Gutman, 1988). Specifically, MEC distinguishes three abstraction levels: attributes – consequences – values (Mulvey et al., 1994). *Attributes* (A) as the least abstract level describe "concrete, physical or observable characteristics" of products. Despite the notion originally describing physical products, the notion can be used for digital products like software, too (Chiu, 2005). *Consequences* (C) constitute the second level of abstraction. They describe what a product provides a user with, either on the positive (benefits) or negative side (costs). A product can have functional or non-functional, e.g., psychosocial, consequences. *Values* (V) are the most abstract level. They represent a user's wishes, goals, and needs and are the end state a customer is trying to achieve through an action (e.g., a purchase). An exemplary ACV chain would be *Spotify (A) – enjoy listening to music (C) – be able to listen to downloaded music on the road (C) – distraction (V)*, as shown in Figure 2.3. A complete ACV chain is commonly referred to as a "ladder" (Russell, Busson, et al., 2004).

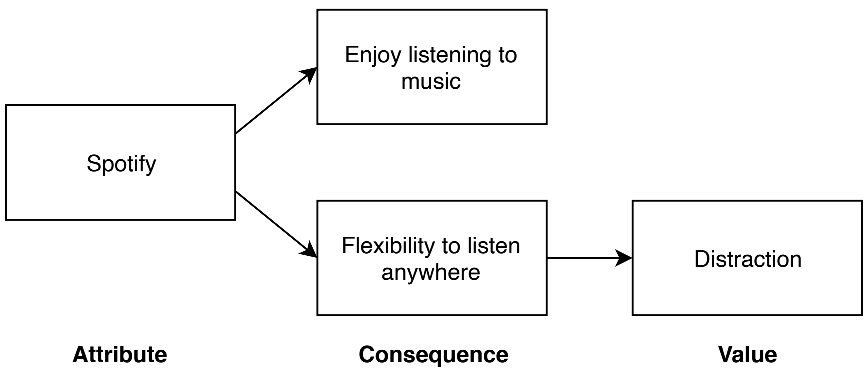


Figure 2.3: Exemplary MEC ladder.

The laddering interview technique can be used to assess such hierarchical structures (Miles & Rowe, 2004). The technique usually includes three steps: 1) Eliciting attributes, 2) performing a laddering interview (including choices about interview structure and techniques), and 3) interview analysis (Jung, 2014). Attributes are the starting point for each ladder. As a seed with a low degree of abstraction, they carry implications for higher-order cognitive processes and determine the direction of the interview (Miles & Rowe, 2004). The laddering interview consists of asking why-questions repeatedly to move between levels of the ACV chain. E.g., an interviewer would start by asking *"Why do you use Spotify? Why is this function important to you?"*. Laddering interviews follow one of two strategies: hard or soft laddering. In hard laddering, participants generate ACV chains one by one, with answers becoming increasingly abstract as participants move from attributes to values. As such, participants stick to one attribute until they complete a ladder (Botschen et al., 2004). On the other hand, soft laddering allows users to jump between multiple attributes, while the actual ladders are only being constructed as part of the analysis (Botschen et al., 2004). Finally, the content analysis of laddering interviews follows a four-step procedure: identifying attributes, consequences, and values amongst the responses; creating a summary matrix by assigning numerical content codes and summarizing all ladders in a matrix; generating an aggregate implication matrix, which contains direct and indirect links between content codes; and finally, visualizing this information in a hierarchical value map (HVM) (Miles & Rowe, 2004). The aim of an HVM is to represent laddering interview data by highlighting dominant connections, whilst still maintaining interpretability (Miles & Rowe, 2004). Figure 2.4 demonstrates one way of visualizing an HVM. For further examples, see Botschen et al. (2004), Chiu (2005), and Jung (2014) or refer to Section 3.2.3.6 for a detailed account of generating a hierarchical goal structure and an HVM.

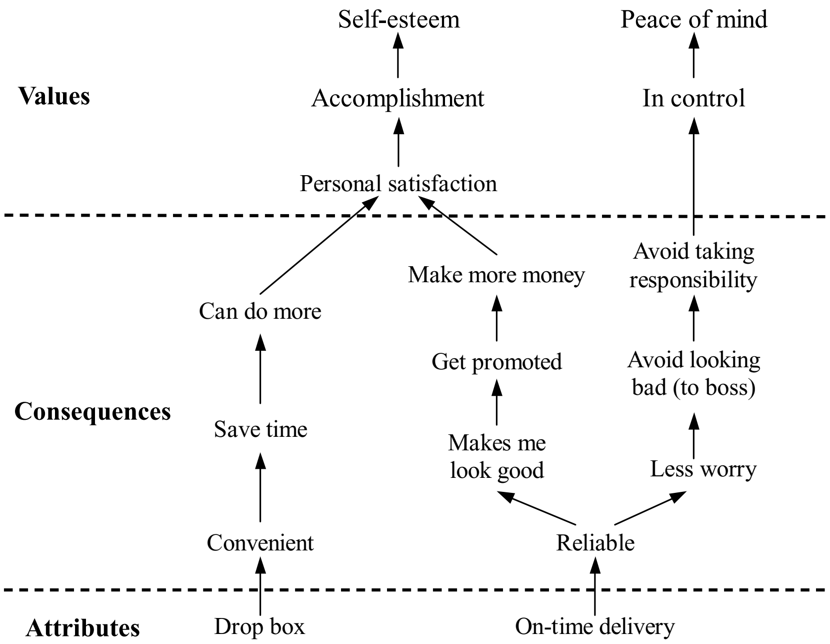


Figure 2.4: Hierarchical value map for an overnight delivery service. HVM example from Chiu (2005).

Laddering interviews have several benefits beyond being a technique for eliciting hierarchical means-end structures. Specifically, (1) laddering interviews are a fairly effective technique for eliciting information, as long as no tacit knowledge is involved (Rugg et al., 2002a; Schultze & Avital, 2011), (2) the information elicited via laddering is structured, which makes it arguably easier to analyze than information obtained from less-structured approaches, such as standard interviews (Gao et al., 2019; Peffers, Gengler, & Tuunanen, 2003b; Peffers, Tuunanen, et al., 2007), (3) the laddering technique provides a streamlined process for analyzing, quantifying, and representing the data (Peffers, Gengler, & Tuunanen, 2003a; Wilhelms et al., 2017).

Traditionally, researchers that desire or require to involve wide audiences in their studies rely on online surveys. Specifically, open questions in online surveys can, to some extent, substitute for interviews – while lacking the interaction between interviewer and interviewee and requiring researchers to develop both structure and questions ex-ante. Since the general structure of laddering, a sequence of why-questions, inherently interacts with participants' answers, thus providing a (minimal) form of interaction, researchers use paper-and-pencil (PP) laddering to engage wide audiences in laddering studies. PP laddering utilizes a questionnaire, firstly asking users for an attribute and then asking, *"which is important to you because..."*, referring to the last response provided. Therefore, PP laddering varies regarding the number of attributes elicited and the number of repetitions of the *which..?* question. While the traditional PP laddering is an offline technique, online questionnaires can be used to increase speed and scalability (Jung & Kang, 2010). Compared with face-to-face interviews, (online) survey-based laddering faces multiple limitations: PP laddering follows a hard laddering approach, usually limited to a predefined scope regarding the number of attributes and responses that are collected (Miles & Rowe, 2004). As users' cognitive structure regarding a topic in question is likely to be more complex, these structures are difficult to capture with a predefined survey (Pieters, Bottschen, et al., 1998). Furthermore, surveys lack ways of interacting and guiding inexperienced users during the interview process, e.g., to overcome mental blockades (Reynolds & Gutman, 1988).

Laddering interviews also come with some disadvantages. Firstly, conducting laddering interviews is time-consuming and costly (Deutsch et al., 2011; Hunter, 1997). This disadvantage is common for qualitative research and not unique to laddering interviews (Miles & Rowe, 2004; Spears & Barki, 2010). Secondly, laddering interviews require highly trained interviewers (Deutsch et al., 2011; Miles & Rowe, 2004) – then again, the training should not be more complex than regular interview training (Bano, Zowghi, & da Rimini, 2018; Kelly et al., 2007). Thirdly, the repetitive structure of laddering interviews may result in participant fatigue and boredom (Kaciak & Cullen, 2009). Relatedly, participants may not be able or willing to answer honestly (Miles & Rowe, 2004). Overall, these factors can restrict data collection from large, representative samples, commonly referred to as wide audiences (Henfridsson & Lindgren, 2010; Tuunanen & Peffers, 2018). Almost every laddering interview study reports sample size as a shortcoming (Gao et al., 2019; Heinze et al., 2017; Jung, 2014; Y. L. Lin & Lin, 2011; Sheng et al., 2005; Wilhelms et al., 2017). Increasing the number of participants could allow the investigation of more groups, ages, and demographics, or enable subgroup analysis (C. F. Lin et al., 2020; Rzepka, 2019).

2.2.2 User Feedback in Online Channels

Researchers have found requirements-relevant information in user feedback on several prominent online channels, including app stores, social media, and product forums (Guzman, Alkadhi, et al., 2017; Pagano & Maalej, 2013; Tizard, Wang, et al., 2019). These channels can contain large volumes of valuable information, as Pagano and Maalej (2013) found that approximately a third of user reviews on app stores contain information related to software requirements. Developers can use user feedback that contains bug reports or feature requests (and more) to address their users' needs and desires, which is critical to their software's ongoing success. Most importantly, channels like app stores and social media are easily accessible for users and IS developers.

However, manually eliciting IS requirements from online feedback can be highly time-intensive due to the large volumes and varying quality of text language from highly distributed user bases (Groen et al., 2017). RE can be further complicated when systems are part of a larger ecosystem, a growing trend in the software landscape (Johnson et al., 2020). In an ecosystem, the line between individual products can be blurred and difficult for users to untangle when giving feedback. Much recent research has investigated methods to automatically extract requirements in user feedback on app stores, Twitter, and product support forums (N. Chen et al., 2014; Guzman, Alkadhi, et al., 2016; Guzman, Ibrahim, et al., 2017; Khan et al., 2019; Maalej & Nabil, 2015; Panichella et al., 2016; Sorbo et al., 2017; Tizard, 2019; Tizard, Wang, et al., 2019).

With the focus on techniques for extracting requirements from qualitative data, there has been limited research aiming to understand the demographics of users that give online feedback and what motivates them. Guzman and Rojas (2019) looked at the difference between women and men who give feedback on the Apple app store. However, the authors manually approximated the gender of each person leaving a written review based on their username. Guzman and Rojas (2019) found a slight majority (57%) of reviews were written by men. There were differences in this ratio when the geographic region was considered. For example, in India, 83% of feedback givers were men. In Australia, women wrote the majority (67%) of the reviews. The authors did not find any statistically significant differences in review sentiment, content, and rating between genders. Another study investigated differences in feedback from the Apple app stores of eight countries (Guzman, Oliveira, et al., 2018). This study found that feedback characteristics such as sentiment, content, rating, and length significantly varied between the countries. However, these studies were both limited to the Apple app store. Also, since demographic information like gender is not available for app store users, both studies could only approximate gender and other demographics, like the feedback givers age, could not be studied.

Overall, recent work has presented evidence that most software users do not give online feedback (Tizard, Rietz, & Blincoe, 2020). Additionally, it showed that certain demographics of users might be underrepresented in feedback, raising questions about how representative online feedback is of the complete user base. In recent years, incentivized crowd-sourced data acquisition has become popular. Platforms like Amazon Mechanical Turk use relatively small financial incentives to elicit crowd-generated data in tasks such as machine learning labeling and research survey's (Buhrmester et al., 2011). Software users are also highly motivated by digital goods, with an excess of $15 billion of in-application spending reported in 2016 (Marder et al., 2019). Previous work has investigated the different motivations users have to acquire digital goods and find game progression, customization, effort expectancy, and social factors to be highly motivating (Bleize & Antheunis, 2019). Further, looking into perceived and actual feedback behavior, a study of Smart Home feedback elicitation found that users reported being enthusiastic to give feedback. However, the actual (real-world) rate of feedback was low (Stade et al., 2020). Additionally, the authors identified that alternative feedback methods such as audio and smart assistant facilitation might encourage feedback compared to traditional methods.

2.3 Qualitative Data Analysis

2.3.1 Coding in Qualitative Data Analysis

The process of qualitative analysis transforms data into findings. There exists a wide range of approaches to the analysis process, such as ethnographic accounts, grounded theory, or content analysis. Their relevance varies based on the research domain and epistemological assumptions of a study (Ritchie & Lewis, 2003). While a comprehensive review of the analysis process and individual approaches is out of scope for this thesis, I refer to Ritchie and Lewis (2003) for in-depth information about qualitative research practices. Furthermore, Sarker et al. (2013) present an excellent overview of qualitative studies in IS.

What unites many approaches to qualitative analysis is that they involve some sort of coding, where researchers aggregate information about the content of data by assigning short labels or *codes* – typically single words, sentences, or paragraphs (Basit, 2003; Evers, 2018; Ganji et al., 2018; Grimmer & Stewart, 2013; Harding, 2015; Lewis et al., 2013; Marathe & Toyama, 2018; Richards, 2002; Wiedemann, 2013). Depending on the epistemological assumptions, researchers take two approaches to coding: *deductive* (codes are derived a priori from scientific theories) or *inductive* (codes emerge from the analytical process). Frequently, coding involves both deduction and induction at different stages of the research process (Patton, 2002; Ritchie & Lewis, 2003). Codes themselves can constitute various levels of information depending on the researcher's needs. Codes are usually created either in a *descriptive* fashion, explaining higher-level concepts, or *in-vivo*, where responses are used directly to create codes and highlight themes. Coding allows researchers to make sense of the vast amounts of data typically created through interviews, field notes, and other qualitative data collection approaches.

The iterative, creative, and human-centered nature of coding (N.-C. Chen, Drouhard, et al., 2018; Richards, 2002) makes it a time-consuming and error-prone task (N.-C. Chen, Kocielnik, et al., 2016; Marathe & Toyama, 2018; Xiao et al., 2020). Code development and application take hours of concentrated work, which is hard to perform reliably at scale (Crowston, Allen, et al., 2012), even for moderately sized datasets. With access to larger datasets and advances in computer-supported analysis, the adoption of qualitative data analysis systems (QDAS) has increased substantially (Evers, 2018; Freitas et al., 2018).

2.3.2 Qualitative Data Analysis Systems

QDAS offer a magnitude of features for organizing, structuring, coding, and analyzing texts and other digital data types such as audio or video to improve upon the traditional paper-based coding procedures (Evers, 2018). Often, the institutional environment determines which systems researchers use due to funding and access to training and support. Prominent examples of QDAS are Nvivo, Atlas.ti, and MAXQDA, with a similar feature set[4].

Despite the importance of coding for the entirety of data analysis, support to accelerate qualitative coding with automated procedures is limited (Marathe & Toyama, 2018).

[4] For a detailed overview of systems and capabilities, see De Almeida et al. (2019) and Freitas et al. (2018)

With recent builds, Nvivo, Atlas.ti, and MAXQDA allow users to search for keywords and auto-code all occurrences (Kalpokaite & Radivojevic, 2018; MAXQDA, 2020; Nvivo, 2020). Nvivo additionally includes an experimental feature that uses machine learning to automatically assign codes using existing coding patterns. The past five years have also seen the rise of various open-source QDAS. *INCEpTION* (Klie et al., 2018) and, more recently, *TEXTANNOTATOR* (Abrami et al., 2019) provide web-based systems specializing in semantic annotation coding. Both systems aim to speed up semantic annotation by integrating active learning from human code examples (INCEpTION) or by providing automated pre-processing of data through named entity recognition, sentiment scores, and topic models (TEXTANNOTATOR). Tietz et al. (2016) specifically evaluate the user interface of their semantic annotation system *refer* which combines manual and automated *annotations* in documents to improve coding quality. They find that a combination of manual and automated annotations achieves the most complete and accurate results (Tietz et al., 2016). As above, the evaluation of user-facing systems so far has focused on enabling users to annotate large-scale datasets for a range of NLP tasks without systematic attention to qualitative data analysis (N.-C. Chen, Drouhard, et al., 2018; Marathe & Toyama, 2018). Focusing on qualitative coding, *Aeonium* uses ML not to speed up coding, but to draw the attention of collaborating qualitative coders to potentially ambiguous data (Drouhard et al., 2017).

Overall, features to accelerate coding in established tools are still at an experimental state and lack transparency, making them hard or sometimes impossible to validate (Grimmer & Stewart, 2013). With a user-centered inquiry, Marathe and Toyama (2018) demonstrate that available QDAS remain "electronic filing cabinets" due to insufficient catering to qualitative researchers' needs. Issues with the quality of and trust in automated code suggestions and a lack of integration in the coding process have led to reluctance in adopting ML-based features (Marathe & Toyama, 2018). Simultaneously, the focus of technologically advanced coding tools lies in supporting corpora creation for NLP tasks. Available systems are not designed to build trust in suggestions through an interactive coding workflow that combines manual and automated annotations (N.-C. Chen, Drouhard, et al., 2018; Marathe & Toyama, 2018).

Applying the MEC approach for research projects by conducting and analyzing laddering interviews has several merits. However, I observe that researchers face common shortcomings and limitations when applying the approach. Firstly, conducting interviews with wide audiences is time-consuming and costly. Secondly, analyzing interview data is tedious and time-consuming – two challenges that are further aggravated with larger sample sizes. Fortunately, the recent advances in AI-supported qualitative research may provide a way for researchers and practitioners to gain access to insights from larger and more diverse samples while increasing the quality and transparency of data analysis.

2.4 AI-based Technology for Qualitative Data Collection and Analysis

2.4.1 AI-based Technology for Qualitative Data Collection

So far, online surveys represented the established baseline for engaging with wide audiences. Recently, researchers looked into automating the interview process using chatbots to circumvent the issues of surveys. However, performing an interview is a complex process and prone to a lack of structure (Yamanaka et al., 2010), insufficient level of abstraction (Moitra et al., 2018), lacking interviewer confidence (Tuunanen & Rossi, 2003), and analyst bias (Appan & Browne, 2012). These faults on the side of the analyst motivate the exploration of supporting activities (Bano, Zowghi, & da Rimini, 2018), such as utilizing a chatbot as a "stable" and controllable interviewer (Nunamaker et al., 2011). As the literature suggests, details of interviews, such as the formulation, ordering, and omission of questions, are crucial, as is the reasoning behavior of analysts (Bano, Zowghi, & da Rimini, 2018). Hence, with the right interviewing technique, a chatbot may be capable of navigating the downfalls of (human) interviewers.

McTear (2002) describes the goal of chatbots as the "[. . .] effortless, spontaneous communication with a computer" (McTear, 2002, p. 2). A systematic literature analysis identified the primary benefits of chatbots to be instant availability, a gentle learning curve, and platform independence (Klopfenstein et al., 2017). Hence, a chatbot should provide the ideal foundation for obtaining information from a wide audience of users. We can differentiate chatbots according to the principles of form and function (Moshagen & Thielsch, 2010; Rinderle & Hoover, 1990). Form characteristics include aspects such as making the bot more human-like in appearance and behavior. Function characteristics strongly influence the utility of a chatbot, e.g. its dialogue control strategy. While a *state-based* bot restricts user input to predefined words or phrases, a *frame-based* bot classifies various questions in multiple "frames". The bot then determines the relevance of a frame according to predefined conditions (McTear, 2002). For example, state-based chatbots can be used to conducting a Likert-scale style survey. The bot asks a series of questions in a predefined order with a stable set of possible user responses (S. Kim, Lee, et al., 2019). Nunamaker et al. (2011) present a frame-based bot that is capable of following distinct paths in an interview script tree based on physiological cues of the interviewee, such as heart rate. The capabilities to react to specific user input provide an advantage over regular web surveys. While modern survey platforms provide ways of reacting to specific responses by adding or omitting certain questions, a chatbot does not require answering questions in a fixed order. Additionally, natural language processing (NLP) capabilities allow bots to react to specific utterances or constellations, triggering predefined questioning techniques (Abdul-Kader & Woods, 2015). Hence, one can equip the chatbot with question techniques used in laddering interviews by human interviewers to assist users when facing difficulties during the interview. Simultaneously, the bot may also apply techniques to "dig deeper" (Reynolds & Gutman, 1988). Recently published studies show promising results of using chatbot interviewers to collect ethnographic data (Tallyn et al., 2018) or gather customer feedback (Xiao et al., 2020). Specifically, chatbot interviewers elicited higher quality responses and encouraged more participant engagement than open questions in surveys (Xiao et al., 2020). However, Xiao et al. (2020) report that their results may be limited by their sample (gamers), the type of questions used, and the rich conversation skills of the particular chatbot used. Researchers have called for studies that evaluate a chatbot interviewer with a different sample (e.g., students), interview strategy (e.g., laddering), and conversation skills (Følstad & Brandtzæg, 2017; S. Kim, Lee, et al., 2019; Rajender Kumar Surana et al., 2019; Xiao et al., 2020). While surveys provide structure to the responses through visual aids and question structuring, chatbots do so through natural conversation (Muresan & Pohl, 2019). Users interacting with a chatbot navigate through a conversation by answering questions one-by-one. Therefore, similar to a face-to-face interview, a bot can react to each response – rephrasing questions in case answers were short or moving to another line of questioning if users do not respond well. Parameters of the chatbot, such as conversation style, as one example of a wide range of social cues, significantly influence how users perceive the bot (Feine et al., 2019; S. Kim, Lee, et al., 2019). The limited number of studies that evaluate chatbots as AI-support for data collection in qualitative research commonly focus on the comparison with established techniques rather than providing a detailed content analysis of the interviews that the bot conducted.

2.4.2 AI-based Technology for Qualitative Data Analysis

One can distinguish two approaches for using ML to support the data analysis of qualitative data from (laddering) interviews: *prescriptive* and *assumptive* (adapted from requirements classification, see Glinz (2007)). In the *prescriptive* approach, ML is utilized by training models for assigning codes to data (Marathe & Toyama, 2018). Most scholars focus on using topic modeling to build interfaces for certain types of qualitative data. Bakharia et al. (2016) evaluate two interactive topic modeling techniques to aid content analysis of open-ended survey questions in a between-subject study, allowing participants to create, merge, and split topics. The authors report that interactivity helped to improve the automatically-generated topics, while trust in the algorithm, on the other hand, was more difficult to improve (Bakharia et al., 2016). More recently, Jipeng et al. (2019) evaluate multiple topic modeling techniques, especially for short texts, by comparing their performance on multiple real-world datasets. Jipeng et al. (2019) conclude that topic modeling provides useful information on document structure, which can help identify the most interesting parts of a document. Furthermore, they call for new ways of visualizing the resulting information to improve how users can utilize it. In the *assumptive* approach, ML is utilized as interactive support that makes suggestions rather than a complete analysis (Glinz, 2005). Marathe and Toyama (2018) compare a search-style query matching technique with two alternative techniques for partially automated coding. They find that this relatively simple technique provides good results, indicating the great potential of ML for interactive analysis support and partial automation. They call for research that designs and evaluates a user-facing interface for partially-automated coding to provide prescriptive (Λ) knowledge (Marathe & Toyama, 2018). In the IS community, N.-C. Chen, Brooks, et al. (2017) presented an interactive tool for analyzing large Twitter datasets. N.-C. Chen, Brooks, et al. (2017) call for more sophisticated tools supporting annotating data on multiple levels of abstraction, based on the real needs of qualitative researchers. Eickhoff and Wieneke (2018) demonstrated the use of topic models in combination with repeated qualitative coding. They support the call for creating tool-support for such approaches and stress the benefits of the assumptive collaboration of ML and manual coding (Eickhoff & Wieneke, 2018).

In the following, I provide a short history of the utilization of prescriptive (ML) and assumptive (NLP) approaches for supporting qualitative coding. Crowston, Liu, et al. (2010) gave a prime example of both approaches by comparing human-created NLP rules against rules inferred with supervised ML. While both approaches offer promise for coding, manual development of NLP rules requires an expert, while ML-based rule development needs many examples. Crowston, Allen, et al. (2012) extended their work focusing on rule-based coding support for content analysis and achieved commendable recall and precision of 74% and 75%, respectively, for some codes. However, creating NLP rules was time-consuming and difficult for rich codes, even for experts that defined rules ex-post from a coded dataset. Meanwhile, the open-source text analysis software *Cassandre* allowed users to define (multiple) single word rules by highlighting *markers* in a text (Lejeune, 2011), which could be grouped under one single label, forming a *register*. Cassandre then gathers all passages that include the marker. Lejeune (2011) referred to the process of iteratively revising markers to improve registers as the *bounce technique*. Shortly after, scholars turned to supervised ML as one way to circumvent the definition of explicit NLP rules and have systems learn directly from manual coding (Grimmer & Stewart, 2013; Lewis et al., 2013). Yan et al. (2014) developed a system for content analysis using a support vector machine and active learning principles for the multi-label classification of emails. While training multiple individual models for each label, they achieved a mean recall of 87% at the expense of precision (7%). Simultaneously, users lacked the technical skills to improve ML models through feature selection and required interactive and adaptive interfaces to understand ML outputs (Yan et al., 2014). Along these lines, N.-C. Chen, Kocielnik, et al. (2016) called for research on interactive ML approaches, reimagining the use of ML in coding to make ML human-understandable. With *Aeonium*, Drouhard et al. (2017) answered the call by giving an example of interactive ML with a system that does not utilize ML to suggest codes but to identify ambiguities. Finally, Marathe and Toyama (2018) reported from an inquiry with qualitative researchers that while researchers desire automation, automation needs to be transparent and part of the coding process. They propose a novel spin at NLP rules by following a search-style querying approach that achieved a commendable 88% precision and 82% recall on average. Compared to the NLP rules used by Crowston, Allen, et al. (2012), search-style rules are more accessible and might force researchers to develop coherent definitions for labels (Grimmer & Stewart, 2013). However, previous work on code rules had experts define rules ex-post rather than following an interactive approach that enabled end-users to define rules as part of the coding process.

Overall, prescriptive ML can perform well for text classification tasks, such as identifying sentiment or modeling topics in unstructured text (Abbasi, 2016). However, ML methods in complex contexts are at risk of lacking domain-specific user input. The assumptive approach, on the other hand, builds on the paradigm of IML. IML places the user in the center of the interaction with the ML system, aiming to create and evolve ML models iteratively through user input, thus creating a good fit to users' goals and needs (Dudley & Kristensson, 2018). This approach enables users to review model outputs, adjust recommendations through feedback, and verify changes. Predominantly, IML is applied for interactive labeling tasks, in which users interact with the system to generate labels for documents, such as images or abstracts (Meza Martínez et al., 2019). Due to its human-centered approach, IML has excellent potential for improving the integration of automation into coding processes by providing transparent and trustworthy recommendations (C.-H. Chen, Trappey, et al., 2016). In the context of coding, the researcher could act as a teacher for the ML model (Knäble et al., 2019). Therein, a researcher interacts with the system in a transparent model development process, where the model learns from iterations of qualitative coding by adjusting coding rules and accepting and rejecting recommendations (N.-C. Chen, Drouhard, et al., 2018; Crowston, Allen, et al., 2012). Existing systems that provide interactive code recommendations build on the ML technique of active learning (AL) rather than IML. AL focuses on identifying new points for labeling by a user to improve the ML model as fast as possible. On the other hand, IML emphasizes the users' role during the process – the user is the driving factor for selecting points to label (Dudley & Kristensson, 2018). In IML, the focus lies on the output of the process (e.g., a high-quality codebook or insights in a qualitative research project), rather than on building an optimal ML model for prediction. For example, INCEpTION integrates active learning to provide annotation assistance and extends the functionalities of WebAnno (Klie et al., 2018). While Klie et al. (2018) give an overview of use cases for AL, they do no perform a structured evaluation of ML-supported coding. Further, INCEpTION focuses on semantic annotation (attaching additional information to concepts, such as people or places) and lacks explanations for recommendations. Aeonium, an ML-based system to draw the attention of multiple coders towards potentially ambiguous data, uses ML to determine which document to label based on predicted ambiguity (Drouhard et al., 2017). Researchers in IS and HCI alike (e.g., N.-C. Chen, Drouhard, et al. (2018), Lindberg (2020), Marathe and Toyama (2018), and Yan et al. (2014)) have called for IML systems to assist qualitative researchers throughout the coding process. However, there seems to be no established design for an IML system for qualitative coding that is grounded in empirical evidence. Furthermore, multiple fields influence the design requirements for such a system, such as IS, Human-Computer Interaction (HCI), Social Science (SS), or Computer Science (CS), complicating the integration of present work. Finally, more research is needed to understand the impact of the interaction with the IML system on users' level of trust (Meza Martínez et al., 2019).

3. Part I: AI-based Qualitative Data Collection & Analysis in IS Development [5]

3.1 Study 1: Ladderbot - A Requirements Self-Elicitation System

3.1.1 Introduction

Digital transformation has brought various information systems into everyone's business and private life, substantially impacting organizations and society (Villela et al., 2018). Literature refers to these changes as a transformation towards a digital society, stressing the influence of the Internet on many traditional services and advocating a power shift towards the user (Leimeister et al., 2014). In the face of persistently high failure rates of ISD projects, it is imperative that an increasing number of users is involved in RE processes, with a varying degree of technological expertise (Jia & Capretz, 2018). The scalable elicitation of user requirements is crucial for developing software that meets needs and demands and reduces project failure rates (Hofmann & Lehner, 2001). Consequently, RE needs to be performed with a wide range of users, who are novices at contributing requirements to development projects (Villela et al., 2018).

For requirements elicitation, interviews have been used most widely (Dieste & Juristo, 2011). Especially the laddering interview is considered a very effective technique for eliciting relevant information for articulating requirements (Dieste & Juristo, 2011). Laddering produces comprehensive and structured insights due to the method's hierarchical nature. In laddering, an interviewer identifies a seed attribute, an initial topic, and asks a series of *why?*-questions to uncover and clarify needs and related attitudes (Miles & Rowe, 2004). While having its roots in personality psychology, laddering has already seen usage for requirements elicitation (Hofmann & Lehner, 2001), e.g., to elicit Customer Attribute Hierarchies (C.-H. Chen, Khoo, et al., 2002). Essentially, requirements are elicited as ACV chains (Miles & Rowe, 2004). Since laddering interviews require highly trained and experienced interviewers, the availability of suitable interviewers imposes a bottleneck onto elicitation interviews (Miles & Rowe, 2004). Tool support is required to enable requirements elicitation with a wide range and number of users (Dieste, Lopez, et al., 2008). Survey-based variants of laddering exist in the form of online and offline *paper-and-pencil* laddering, increasing the scale of the technique independent of the need for interviewers. However, this method faces multiple limitations: It restricts interviewees' responses (Pieters, Bottschen, et al., 1998; Russell, Busson, et al., 2004), provides little assistance in the case of misunderstandings or problems (Miles & Rowe, 2004), and fosters boredom and fatigue due to a repetitive question structure (Kaciak & Cullen, 2009). One needs to understand the characteristics of novices' requirements (self-)elicitation behavior to understand the implications for a novice-centric self-elicitation system. In this thesis, I use the term *self-elicitation* to describe the process of users interacting with a system to produce requirements-related qualitative data. As the user is guided in uncovering

[5] This chapter is based on the following studies which are published or in work: Rietz and Maedche (2019), Rietz and Maedche (2021b), Rietz and Maedche (2021a).

their requirements, rather than being enabled to create a service with a direct benefit for themselves, I argue that self-elicitation serves as a better term than *self-service RE system* to describe the process. This study aggregates common challenges of RE interviews with novice users and presents the design and the architecture of a laddering chatbot for interviewing novice users: Ladderbot.

Several tools have been proposed over the years to aid with RE. Derrick et al. (2013) evaluated an embodied conversational agent to facilitate a group workshop that used prompts to guide and assist during user story formulation. AnnotatePro allows users to submit requirements that can be drawn on their screens (Rashid et al., 2006). These approaches, amongst others such as WinWin (Boehm et al., 1998) or EasyWinWin (Grünbacher & Boehm, 2001), allow users to communicate requirements. However, these tools do not consider users' particular level of experience, limiting the utility of such tools for novice users. Tools such as FAME (Oriol et al., 2018) and ASSERT (Moitra et al., 2018) cater to novices, but only on the side of novice analysts, not novice users, hence not enabling self-elicitation. Guidance and assistance are necessary to elicit high-quality requirements from novice users (Kato et al., 2001; Mohedas et al., 2015). Ladderbot tackles these challenges by utilizing questioning techniques adapted from guidelines for laddering interviews (Reynolds & Gutman, 1988). At the same time, building on established interviewing techniques enables Ladderbot to collect information about users' cognitive structures (Russell, Busson, et al., 2004) that goes beyond the capabilities of PP laddering.

3.1.2 Designing a Laddering Interview Chatbot for RE

3.1.2.1 Common Issues of RE Interviews with Novice Users

When involving wide audiences into RE, it is reasonable to expect that many users are novices at contributing requirements to ISD projects. Understanding the specific challenges that arise from conducting RE interviews with novice users is essential. In the following, I present prescriptive knowledge aggregated from relevant literature on supporting novices during RE interviews, mostly from guidelines for novice RE analysts (e.g., the interviewers in RE interviews).

Overall, the relevant literature rarely focuses on supporting novice users during RE (Villela et al., 2018). Moreover, novice RE analysts are the focus of supporting activities (Bano, Zowghi, Ferrari, et al., 2018). Insights from analyzing novice analysts' behavior in elicitation processes may serve as a guideline for providing appropriate support for users in contributing requirements to development projects. One of the most frequently observed downfalls during elicitation processes with novice users or novice analysts performing elicitation is a lack of structure (Yamanaka et al., 2010). Experienced interviewers utilize business and domain knowledge to inform the questioning structure and follow-up questions. As novices cannot rely on such prior knowledge, utilizing a fixed interview structure can help achieve a consistent elicitation quality (Yamanaka et al., 2010). A lack of interview structure is frequently reflected by interviewers not digging deep enough when conducting interviews, which impacts the correctness and completeness of requirements-relevant information (Kato et al., 2001). Specifically, novice interviewers tend not to ask enough *why?*-questions. It is vital to ask why?-questions to understand reasons for a demand, sources of a need, and ultimately, the values that a user aims to achieve. Without asking appropriate follow-up questions, interviews are at risk of remaining shallow, and time is spent with unnecessary questions. Novice users, in particular, are not familiar with communicating requirements, which may be rooted in an incomplete understanding of their own needs. Thus, the task of uncovering the cause of a need or requirement falls to the interviewer. Otherwise, interviews lead to ambiguous user statements at the wrong level of abstraction (Moitra et al., 2018). Without uncovering the cause of or foundation for user needs, the development of disruptive solutions stagnates. Furthermore, novice analysts make procedural mistakes during interviews, such as formulating questions wrongly (e.g., biasing interviewees through leading questions), ordering questions incorrectly (e.g., no attempt of having a good start or end of the interview, or asking questions in an incorrect logical order), and question omission (e.g., no probing questions). A predefined interview structure can help avoid such mistakes (Bano, Zowghi, Ferrari, et al., 2018). Besides explicit mistakes during performing an elicitation interview, implicit aspects such as an interviewer's behavior substantially impact the results of an interview. Specifically, interviewers may (unconsciously) display a lack of confidence, lack of professionalism, or have inadequate time management (Bano, Zowghi, Ferrari, et al., 2018). Such behavior can impact users' attitudes and influence their responses, with overconfidence of the interviewer being especially dangerous in potentially leading to an incorrect understanding of the problem domain. Finally, experienced interviewers commonly think in and explore a problem domain in relationships between concrete attributes and underlying implications, desires, and values. They utilize this view based on a model of relations to guide their reasoning and interview behavior (model-based reasoning behavior) (Sutcliffe & Maiden, 1992). On the other hand, novice interviewers often cling to surface similarities between information in their reasoning and fail to explore first-glance similarities in more detail, e.g., through abstraction and analogies (reasoning behavior based on object-attribute similarity). Here, visualizing relationships between concepts can help novice interviewers create associations and potentially identify hasty conclusions (I.-L. Huang & Burns, 2000).

Both structural and behavioral interview guidelines for novice interviewers are necessary for eliciting high-quality requirements. Table 3.1 summarizes these guidelines and contrasts the summary with the benefits and difficulties of laddering interviews (see Section 2.2.1) and the perks of chatbots (see Section 2.4.1). Overall, the combination of laddering interviews with a chatbot interviewer shows promise as means to perform RE interviews. Laddering interviews provide a fixed structure based on a series of why?-questions that explores requirements in multiple levels of abstraction. Additionally, its hierarchical nature supports the identification of relations between concepts (Reynolds & Gutman, 1988). Difficulties regarding interviewer bias and the time investment for performing interviews may be offset through chatbots as automated interviewers. In the following, I present the design and the architecture of Ladderbot as an instantiation of a laddering chatbot.

How to support novice end-users?		Source
	As requirements elicitation is influenced by individual business knowledge and experiences, elicitation quality differs. Consistent quality can be achieved by utilizing a **fixed structure** for requirements elicitation work.	Yamanaka et al. (2010)
	Interviews tend to not dig deep enough, asking not enough why?-questions. Hence, requirements are not elicited correctly and time is spent with unnecessary questions. Using a **set of why?**-questions can be useful for uncovering the underlying values of users.	Kato et al. (2001)
	Novices formulate ambiguous statements at an insufficient level of abstraction. Lack of information complicates requirements conflict resolving. **Explaining a requirement on multiple levels of abstraction** can increase the level of detail.	Moitra et al. (2018)
	Expert information analysts use a model-based reasoning behavior, while novices rely on object-attribute mapping. **Visualizing relationships between concepts** may help novices to create associations.	I.-L. Huang and Burns (2000)
	Technical and soft skills for conducting interviews require practice. **Question formulation, ordering, omission** as well as the **behavior** of a (virtual) interviewer can be controlled for a virtual agent.	Bano, Zowghi, Ferrari, et al. (2018)

Perks of chatbots		Source
	Effortless, barrier-free interaction	Klopfenstein et al. (2017)
	State-based **dialogue guidance**	McTear (2002)

Benefits and difficulties of laddering interviews		Source
Benefits	Can clarify requirements	Mulvey et al. (1994)
	Hierarchical nature allows good understandability Parts of ACV chains can be reused	Jung (2014)
	Structured information Effective mode for eliciting information	Miles and Rowe (2004)
Difficulties	Requires ability to express knowledge of a domain and structure it hierarchically	Chiu (2005)
	Long and tiring technique	Dieste and Juristo (2011)
	Interviews are time-consuming / costly Requires highly trained interviewers Potential interviewer bias Potential for fatigue and boredom	Miles and Rowe (2004)

Table 3.1: Overview of the conceptual foundations of Ladderbot.

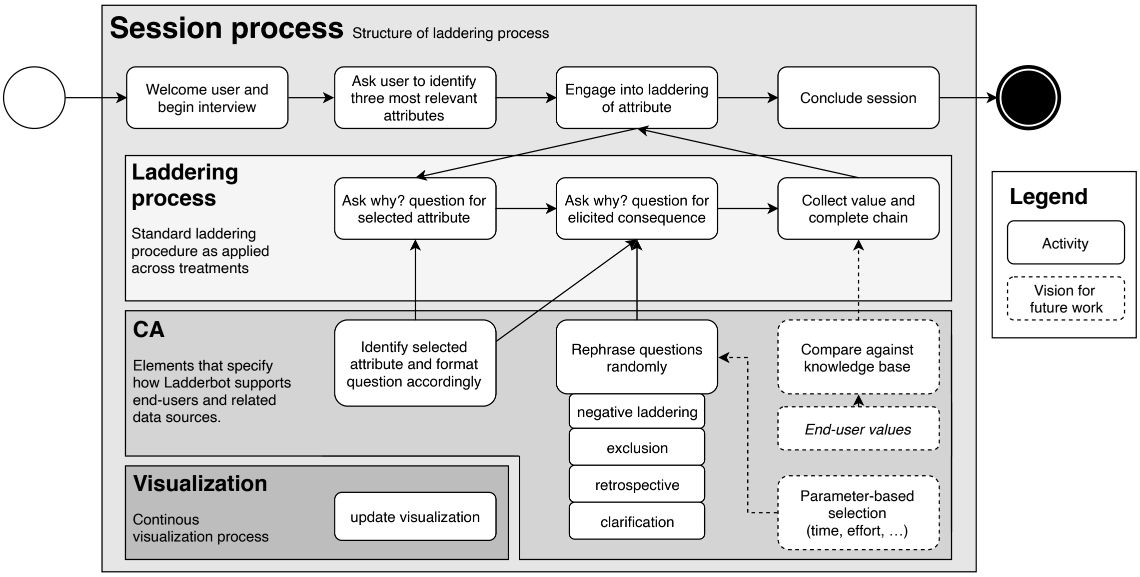


Figure 3.1: Activity map of Ladderbot.

3.1.2.2 Chatbot Structure for Laddering Interviews

The frame-based laddering chatbot uses a predefined set of questions manipulated during runtime to appear more human-like. I built the chatbot using the Microsoft Bot Framework (SDK 4) for Javascript. Ladderbot follows a hard laddering approach that requires users to complete a ladder for one attribute before changing to the next. During the interview, the chatbot switches between three dialogues, which control three segments of the interview: the elicitation of three attributes, a series of questions to elicit a ladder for each attribute, and a control dialogue to initiate each ladder and conclude the interview process. The way the chatbot performs the steps of attribute elicitation, ladder introduction, and interview conclusion is identical for every user. Figure 3.1 provides an overview of the activities of Ladderbot.

Technique	**Description**	**Example**
Negative laddering	Ask the user why they *do not* do something or *do not* want to feel a certain way	Why would you not apply for a job where overtime work is not tracked?
Exclusion	Ask the user to imagine a situation where an attribute or consequence does not exist	What would you base your decision on if you could not choose an employer with over 100 employees?
Retrospective	Ask the user to imagine their behavior in the past and compare it to now	Compared to a couple of years ago, have your preferences changed?
Clarification	Repeat the user's response and ask for clarification	Please allow me to clarify. You said that 'You want to make a lot of money with your education. So, why is that important to you?

Table 3.2: Guiding techniques used by Ladderbot.

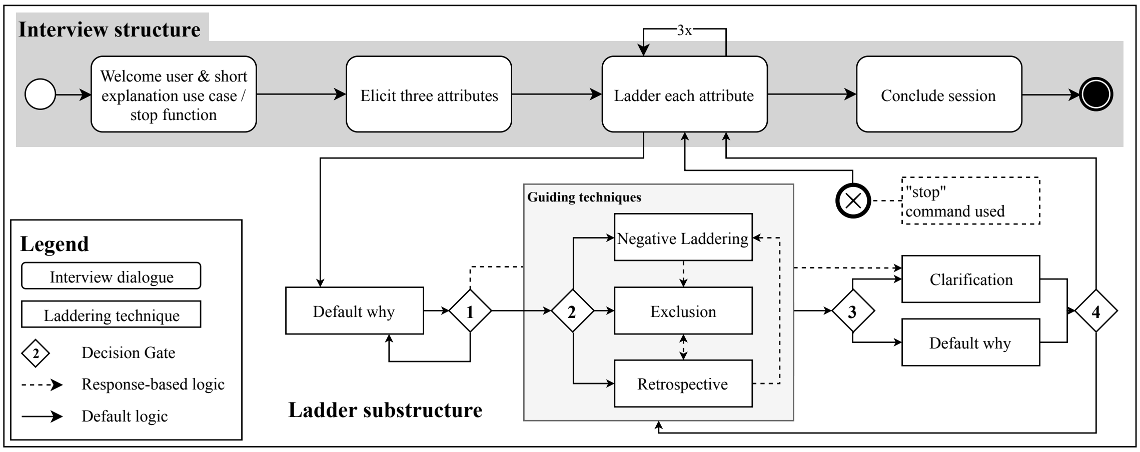


Figure 3.2: Interview structure of Ladderbot. The question selection is controlled with four decision gates.

The central part of a laddering interview - laddering each attribute - is semi-structured, and the exact sequence of questions varies between participants. In this part, the chatbot uses 29 question variants and prompts to conduct the interview and react to certain utterances and responses. Most importantly, the chatbot uses four guiding techniques (negative laddering, exclusion, retrospective, and clarification, with five possible variants for each) combined with variants of the default why-question (five variants). Table 3.2 shows examples for each guiding technique. I adapted the questioning techniques from Reynolds and Gutman (1988). Furthermore, four special questions and prompts guide the interviewee, e.g., an introduction, an explanation of the *stop* (that allows users to end the questioning sequence for the current attribute and switch to the next one), or an end message. During the interview, the chatbot follows a strategy of rule-based randomization to come up with the next question. This rule-based randomization is implemented by including four decision gates into Ladderbot's decision logic. At each decision gate, Ladderbot decides which question to ask next according to weights assigned to each gate as illustrated in Figure 3.2. These gates follow a set of rules, as outlined in Table 3.3. The decision gate control structure constitutes the default structure of each laddering interview. Additionally, Ladderbot reacts to predefined responses and adapts its structure accordingly (shown as dashed arrows in Figure 3.2). Response-based reactions are primarily triggered by responses that negate a question.

Ladderbot: *Do you think that the function Whatsapp could cause problems? Can you think of solutions to these problems?* User: *No.* Ladderbot: (reacts to the negative response) *Did anything ever bother you about the Whatsapp function or did something not work?*

For specific questions, Ladderbot also pays attention to the length of a response. For example, when Ladderbot asks a user whether they would have answered differently in the past (retrospective) and the user replies with *Yes*, but with no further details, Ladderbot would ask a follow-up question. A benefit of this dialogue control structure is that no special domain knowledge is necessary to configure Ladderbot. The 29 questions and prompts currently used are generic laddering questions that depend more on the structure of the interview language than on domain knowledge. As such, Ladderbot can be reconfigured for

Gate	Rule
1	Determines the number of default why-questions before using guiding techniques. The bot will initiate the interview by asking, "*Your 1. example was *attribute*. Why do you use *attribute*? Which benefit does this function provide to you?"*. A new default why-question is asked with a likelihood of 75%. If so, the weight is reduced by 0.5, and Ladderbot remains at the gate. Consequently, there will be two additional default why-questions max. before the bot begins using guiding techniques.
2	Gate two determines the likelihood of each guiding technique. Weights are initiated to be 1/3 each, giving each technique the same likelihood. Once a technique has been used, the weight for this technique is set to 0. Hence, each guiding technique may only be used once per ladder.
3	Gate three decides if the default why question or the clarification technique should be used to ask a follow-up question to the previous guiding question. The weight at gate 3 remains constant at 0.5 during the interview, giving both question techniques the same likelihood.
4	Gate four is used to end the current ladder and is instantiated with 0. The weight is set to 1 if every guiding technique has been used during a ladder to have Ladderbot switch to the next attribute.

Table 3.3: Decision gate control structure.

multiple use cases with minimal effort. No variant of a questioning technique is used twice during an interview. For the default question, no variant is used twice while laddering an attribute. Interviewees are required to answer at least three questions per attribute. Subsequently, interviewees can tell the chatbot to continue with the next ladder by typing "stop." In case no attribute is left to ladder, the chatbot ends the interview. Ladderbot does not impose restrictions on the length of a response to keep the interaction natural. Users are not capable of editing given responses. Gates are instantiated with the following weights for each ladder: (1) 0.75 | (2) 0.33, 0.33, 0.33 | (3) 0.5 | (4) 0. I selected these weights to have Ladderbot ask up to three why-questions before using guiding techniques, similar to a PP survey. Appendix A.1 shows an complete interview with Ladderbot.

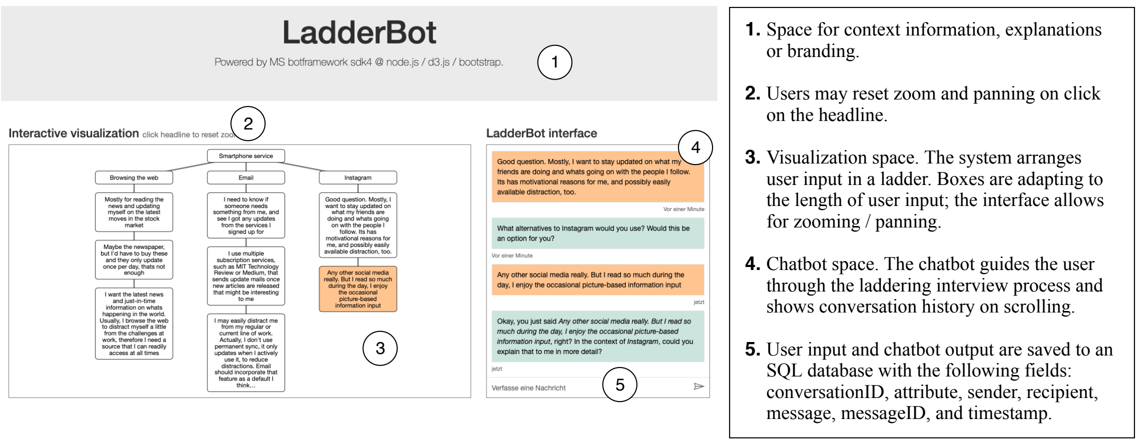


Figure 3.3: Overview and explanation of the Ladderbot user interface.

3.1.2.3 Interview Visualization

Each user response is visualized next to the chatbot in a tree structure, using d3.js. This visualization of the interview history and structure resulted from one of the first pretests I conducted using a think-out-load approach. While testing Ladderbot, one of the testers stated, "*[...] using a questionnaire, I can see what I said so far, what answers I gave [...]. With a chatbot, I would have to scroll through the entire conversation to see if I gave that answer already. I lose the overview*". As such, I implemented a visualization that adds every user's response into a tree diagram, visualizing the steps of the laddering interview. Eventually, the tree forms three branches during a completed interview (one for each attribute). Ladderbot's questions are omitted from this visualization to focus the user on their argumentation structure solely. Users can zoom and pan the visualization to look at individual branches in detail, as the tree tends to become small as it fills throughout the interview. Figure 3.3 demonstrates the user interface of Ladderbot and explains its five main sections.

3.1.3 Conclusion

Study I presents the design and architecture of Ladderbot, a requirements self-elicitation system capable of conducting domain-agnostic laddering interviews with novice users. Ladderbot guides the interviewee to generate attribute-consequence-value chains in three steps. First, the bot asks users for their favorite attributes for a use case. Second, the bot asks a series of guiding questions to elicit a ladder for the first attribute that interviewees mentioned. Third, the bot repeats the questioning structure for attributes two and three. During the interview, the bot continuously updates a dynamic visualization of the interviewee's answers. Elicited attributes, consequences, and values are visualized for the user in a tree-like shape throughout the interview process. The graphical representation of ACV chains may assist users in structuring their thoughts and uncovering new relations (I.-L. Huang & Burns, 2000). So far, several comparisons of elicitation techniques have identified laddering as a powerful technique. However, only a limited amount of research describes approaches to creating tool support for laddering, especially for tool-supported self-elicitation of user requirements. For example, Jung (2014) applies a combination of regular laddering interviews and PP laddering to identify user values of smartphone usage while investigating the means-end chain approach in the context of IT-user studies. However, no approaches are presented to assist the interviewer or to completely automate the interview process. This is also true for Tuunanen and Rossi (2004), who propose a method for broad-based requirements elicitation that requires human-led interviews for ACV chain generation. A similar approach to the idea of chatbot-based laddering interviews was presented from Kassel and Malloy (2003), who attempt to automate requirements elicitation through combining domain knowledge, a software requirements specification (SRS) template, and user needs as XML in a tool-based approach. However, their approach uses closed-ended questions, whereas the laddering tool proposed in Study I pays particular attention to those details that are introduced through open-ended questions.

Overall, we expect Ladderbot to allow the elicitation of requirements from users without the need for highly qualified interviewers. Furthermore, enabling users to self-elicit requirements creates the potential to contact a broader range of users, hopefully improving software development projects through detailed insights. In the spirit of ”RE for everyone” (Villela et al., 2018), tool support for users enables developers to get an idea of the expectations of society and supports the end-to-end value co-creation between an outer- and an inner circle of systems development teams: between users and system engineers, analysts and developers.

3.2 Study 2: Re-Evaluating User Values of Smartphones - A Wide Audience Qualitative Research Study

3.2.1 Introduction

In 2014, Jung published a study with twofold intentions: First, from a topic perspective, he described a set of user values of smartphones that his empirical work has discovered. Second, from a methodological perspective, he introduced the MEC approach into IS usage research (Jung, 2014). The MEC approach allowed Jung to investigate the relations among values and user goals, thus providing a richer picture of why and how individuals use smartphones. Jung calls attention to the conceptual shift in IS research towards studying what users do with technology, respectively, the goals and values users pursue with the technology (i.e., value-oriented perspective). Thereby, the value-oriented perspective complements studies focusing on factors affecting user adoption of a given information technology (i.e., the user adoption perspective) (Jung, 2014). Jung's study also advanced value-oriented research by refining and expanding the abstract values (e.g., utilitarian, hedonic, and monetary) used in prior research. While Jung's study was novel in its application of the value-oriented perspective to IS usage research, related approaches can be found in earlier IS journal papers. Sheng et al. (2005) apply value-focused thinking for identifying values of mobile technology for an organization, as well as relationships between these values. However, Jung's study stood out amongst value-oriented studies in IS regarding the number of involved interviewees (n=54). Rarely did studies involve such a large number of participants (e.g., Y. L. Lin and Lin (2011)), due to the time and costs involved in facilitating a large number of interviews and analyzing corresponding vast amounts of data (Deutsch et al., 2011).

Despite its strengths, the means-end approach has only gained minor adoption in IS journals and conferences (Chiu et al., 2014; Jung, 2014; Rzepka, 2019; Tuunanen & Kuo, 2015). Firstly, MECs provide a structure for quantifying and analyzing qualitative data, which is extremely helpful for comparing multiple interviews (Rugg et al., 2002a; Wilhelms et al., 2017). Secondly, they allow for a detailed analysis of usage motives and cognitive motive structures (Wilhelms et al., 2017). Thirdly, they help explain the relationships among goals due to their inherent hierarchical structure (Jung, 2014). Fourthly, they align with a broader trend in the IS and other disciplines, focusing on values as a source for a deeper understanding of outcomes and lasting impacts (Van Mechelen et al., 2017). Given these manifold benefits of MECs, what could explain its rarity in IS research? MEC studies predominantly present similar shortcomings of their work: limited sample sizes and homogeneity in age, demographics, and the participant sample as a whole (Gao et al., 2019; Heinze et al., 2017; Jung, 2014; C. F. Lin et al., 2020; Rzepka, 2019; Wilhelms et al., 2017). Furthermore, MEC studies are intensive in terms of time and personal requirements because the primary method for data collection are interviews which lack scalability to wide audiences (Deutsch et al., 2011; Miles & Rowe, 2004).

Study II pursues two objectives to address these challenges. Firstly, I provide an update to Jung's investigation of user values of smartphones. As information technology, including smartphones, is evolving rapidly, I was interested to see how user perspective on smartphones has changed over the last years. Additionally, recent research on smartphone usage mainly focuses on negative gains, such as addiction or loss of privacy (Keith et al., 2015; Sutanto et al., 2013; Vaghefi et al., 2017). Jung's original study excluded negative gains due to the MEC approach's original focus on means to achieve positive goals (Jung, 2014). This study explores both positive and negative gains and values. Secondly, I use artificial intelligence (AI)-based technology to conduct the study with a large participant sample. Recent advances in AI-based technology in the fields of natural language processing and machine learning provide means to enable reaching out to a wide audience in qualitative research. In particular, text-based conversational agents, so-called chatbots, have shown promise as a tool for interviewing without supervision (S. Kim, Lee, et al., 2019; Nunamaker et al., 2011; Tallyn et al., 2018). This study combines laddering interviews, the prevalent interviewing techniques used in the MEC approach (Reynolds & Gutman, 1988), with a chatbot interviewer, to perform qualitative research with a large sample. As a baseline for evaluating chatbot-based laddering, I use two variations of online surveys that build upon the state-of-the-art approach for conducting laddering interviews with wide audiences: the PP laddering questionnaire. Therein, I compare the benefits and shortcomings of online surveys and chatbots as means to perform laddering interviews and contrast my findings with the results of Jung's and other more recent studies (Hedman et al., 2019; J. Park & Han, 2018). I present the results of my evaluation to understand how user values in smartphones have changed since Jung's study. Additionally, I present the comparison of the chatbot interviewer's results with the two survey-based approaches, based on behavioral and perceptional constructs.

My novel contributions include the following: First, I present an updated perspective on smartphone user values based on qualitative data collected from a wider audience. Second, I apply and compare state-of-the-art methods with a laddering chatbot in the context of conducting interviews with wide audiences. In online survey- and chatbot-based interviews with 256 participants, conducted over one week, I find that smartphones are predominantly means to communicate and achieve socialization. Secondly, smartphones allow users to pursue intellectual and emotional self-optimization towards the end of satisfaction. Interestingly, smartphone users prioritize social or utilitarian values over convenience, which has implications for practitioners competing in the increasingly commoditized and free-to-use market for smartphone apps. Furthermore, I identify the negative impacts of smartphones. I find that users are wary of how smartphones promote and force behavioral change, particularly regarding communication. Finally, I discuss the strengths and weaknesses of online surveys and chatbots for wide audience involvement. Survey-based laddering more reliably produces ladders that end in values, while my approach to chatbot-based laddering sacrifices some structure to explore negative gains. However, the chatbot engages participants to give significantly more and longer answers and guides participants during the interview process, resulting in significantly higher learnability. I conclude by presenting implications for value-oriented research and strategies for wide audience laddering interviews. Additionally, I discuss implications for tech companies to inform development and marketing decisions and highlight the value of supporting smartphone usage in the workplace.

3.2.2 Background

3.2.2.1 Value-Oriented Research

Value-oriented research has a long history in IS research, however, the number of published studies is small, as shown in Table 3.4[6]. Further, the used terminology varies between studies: While Jung uses the term value-oriented approach to highlight the focus on goals or values that users pursue with technology, other authors referred to value-focused thinking (Heim et al., 2009; Nah et al., 2005; Sheng et al., 2005), personal construct theory (Peffers & Gengler, 2003), or (personal) values of individuals (Bourne & Jenkins, 2005; Kuisma et al., 2007; Sun et al., 2009). More recently, researchers refer to value-oriented research as value-focused thinking (Gao et al., 2019; Rzepka, 2019) or value-based view (Heinze et al., 2017; Tuunanen & Kuo, 2015). The value direction of research is appreciated among practitioners for its approaches to not only identify values but to structure identified values and relationships systematically (Gao et al., 2019). Further, the inherent focus on outcomes and lasting impacts fit a broader trend in related disciplines, e.g., human-computer interaction (Van Mechelen et al., 2017). I used the publication date of Jung's article as a landmark to separate studies published in IS outlets that follow a value-oriented approach into two groups (pre-2014 and 2014-today). While the number of studies published remained similar, key characteristics changed: the average number of participants increased from 32 to 45. The average duration of interviews decreased from 53 to 35 minutes (amongst papers that report the number of participants and duration of interviews). Overall, the time that researchers spend interviewing participants decreased only slightly from 28.5 to 26.5 hours. Regardless, researchers spend considerable time collecting data, often followed by a tedious, error-prone, and overwhelming data analysis process (Abbasi, 2016; N.-C. Chen, Drouhard, et al., 2018; Tuunanen & Kuo, 2015).

3.2.2.2 User Perspectives on Smartphone Usage

Smartphones have become an extension of their users and are interwoven into many aspects of everyday life. This inspired various studies over the past decade to look into smartphone usage, both in and outside core IS outlets (Bødker et al., 2014). Before and around the early 2010s, user adoption was the most popular research theme in IS (Ladd et al., 2010). Particularly the technology acceptance model (TAM) was used to analyze hedonic and utilitarian intrinsic values and social influence in device usage (Chun et al., 2012; Wakefield & Whitten, 2006), or the adoption among individual demographics (D. Kim et al., 2014) or professions (Y. Park & Chen, 2007). Furthermore, researchers started to investigate individual values, such as personalization and privacy (Sutanto et al., 2013) or aesthetics (Shin & Choo, 2012) to understand better how different value preferences influenced usage intentions and smartphone perception. To that end, IS researchers began to study the effects of smartphones with longitudinal usage studies. Usage of smartphones was shown

[6] I identified the presented articles in a systematic literature search with the search string *(("means end") OR ("means-end") OR (laddering)) AND value** in the Scopus database in October 2020. I selected studies published in AIS Basket Journals, SIG Recommended Journals, and AIS conferences. Of 1016 initial hits, 47 studies were published in the mentioned outlets, of which 24 featured value-oriented research studies that use laddering.

Source	Topic	Method
Jolly et al. (1988)	Cognitive bases of performance appraisal	Interviews lasting from 1-2 hours with 22 nurse supervisors
Peffers, Gengler, and Tuunanen (2003a)	Facilitate broadly participative information systems planning	Two interview case studies lasting on average 40-50 min. with 32 participants
Bourne and Jenkins (2005)	Managers' personal values	20-30 min. laddering interviews with 7 senior managers
Sheng et al. (2005)	Strategic implications of mobile technology	Interviews lasting 30-45 min. with 12 sales representatives and district managers
Chiu (2005)	Elicit system requirements and understand users' perceptual orientations	4 focus groups with weekly 1.5 hours meetings for one month with 8 members each from university staff and part-time graduate students
Sheng et al. (2005)	Value of mobile applications in a utility company	Interviews lasting approx. 1 hour with 10 employees
Kuisma et al. (2007)	Resistance to internet banking	Interviews with undisclosed length with 30 ATM customers
Heim et al. (2009)	Customer value of RFID in service applications	Qualitative survey responses of 101 undergraduate students
Sun et al. (2009)	Critical functionalities of successful e-learning systems	(Virtual) telephone interviews lasting 40-60 min. with 31 instructors
Y. L. Lin and Lin (2011)	Goal values for MMORPG players	Interviews lasting 45-60 min. with 60 players
Yang and Chang (2012)	Customer's decision process in selecting bundles	Interviews with undisclosed length with 48 cosmetic experts
Pai and Arnott (2013)	User adoption of social networking sites	Interviews lasting 30-45 min. with 24 Facebook users
Jung (2014)	Understanding user values of smartphones	Interviews lasting 30 min. on average with 54 undergraduate students
Jung (2014)	Goal structures of consumers in social virtual worlds	Text-chat interviews lasting 20 min. on average with 93 Second Life users
Zaman et al. (2014)	Motivation profiles of online poker players	Interviews lasting 50 min. on average with 18 young adults
H. W. Lin and Lin (2014)	Digital educational game value hierarchy from a learner's perspective	Interviews lasting 45-60 min. with 50 SimCity players
Y. L. Lin, Lin, and Hung (2015)	Target values of learners in massive open online courses	VoIP Interviews lasting 45-60 min. with 60 learners
Tuunanen and Kuo (2015)	Value-based view of requirements prioritization	Interviews with undisclosed length with 83 lead users
Wilhelms et al. (2017)	Peer-providers' participation motives in peer-to-peer carsharing	Interviews with undisclosed length with 20 P2P carsharing members
Heinze et al. (2017)	Customer resistance to mobile commerce of insurances	Interviews lasting 36 min. with 23 consumers
T. H. Huang et al. (2018)	Customers values from brand fan pages	Interviews with undisclosed length with 35 students and office workers
Gao et al. (2019)	Value of smartphones for older adults in China	Interviews with undisclosed length with 11 old adults
Rzepka (2019)	Value of voice assistants	Interviews lasting 25 min. on average with 31 voice assistant users
C. F. Lin et al. (2020)	Young people's perceptions of social networking sites	Interviews lasting 50 min. on average with 62 young Taiwanese

Table 3.4: Value-oriented research using the laddering technique in IS outlets.

to have become ubiquitous, in that it can be distinguished in conscious (i.e., time-out) and unconscious (i.e., time-in) use (Bødker et al., 2014). As smartphones are constantly in use both in private and professional use cases, disconnecting from work is neither easily possible nor desirable for many users (Dery et al., 2014). As such, studies began to touch on the negative gains of smartphone usage and the trade-offs and negative values (guilt, anxiety) that can come with smartphone usage (Dery et al., 2014). At the same time, a longitudinal study outside the IS domain presented a taxonomy of values achieved with smartphone use as a subset of a large category of life values, which are defined as desirable states of existence or modes of behavior (J. Park & Han, 2013). J. Park and Han (2013) present fifteen user value elements to help understand what users seek to achieve with their smartphones, including *convenience*, *pleasure*, *beauty*, and *friendship*. In 2018, Park and Han expanded on the case of smartphones for evaluating prototypes that were created through value-centered design. Park and Han found the most used smartphone attributes to be texting (using third-party applications, e.g., Whatsapp), social network services, and calls. However, the participants in the study rate other attributes as most valuable, with the camera, voice recording, and weather applications achieving the highest scores.

Meanwhile, IS research began to investigate some of the negative outcomes of heavy smartphone usage, focusing on excessive use, IT addiction, and privacy concerns. Research on smartphone-induced IT addiction studied the effects of demographics (Kwon et al., 2016), different addiction types based on individual liability to addiction (Kuem et al., 2020; Vaghefi et al., 2017), problematic smartphone game use (C. Chen et al., 2020), and compulsive use (Bødker et al., 2014), specifically of social network services (C. Wang & Lee, 2020). Smartphone privacy research has focused on information disclosure via mobile social apps (Kwon et al., 2016) and via device-specific functionality, e.g., location tracking (Crossler & Bélanger, 2019). While the overall focus of smartphone-related research has shifted towards some of the negative outcomes of IT adoption (Vaghefi et al., 2017), recent studies were concerned primarily with a top-down investigation of (mental) health-related outcomes, such as problematic use and addiction or with supporting well-being (Stawarz et al., 2019). However, bottom-up studies probing individual users for both positive and negative gains of their smartphone usage remain scarce. Outside the core IS outlets, one can observe a similar shift away from pure adoption-related research towards problematic smartphone use, although not with the same intensity. Researchers remain occupied with investigating factors influencing behavioral intentions of non-smartphone users (C. Y. Lin et al., 2017) and the influence of lifestyle clusters on usage intention (J. H. Kim et al., 2018). Further, health became a major topic for smartphone-related research (Pedrero-Pérez et al., 2019; Richardson et al., 2018; Stawarz et al., 2019). The health-related stream brought forth several studies not directly related to smartphone-usage, that evaluated smartphones as a digital health system and alternative to conventional and specialized devices. This includes health monitoring (Nemcova et al., 2020), support for persons with disabilities (S. Kim, Chang, et al., 2020), or hearing aids (Ho et al., 2020).

While this short review highlights the exciting and relevant smartphone-related research that has been published over the last decade, both inside and outside core IS outlets, it also reveals several gaps. *Firstly*, research is commonly focused on negative implications of smartphone usage, such as addition and privacy concerns, but rarely investigates negative gains directly with smartphone users (Jung, 2014; J. Park & Han, 2018). *Secondly*, value-oriented studies in the past ten years on average include less than 50 participants, and scholars have called multiple times for studies with larger sample sizes (Gao et al., 2019; Heinze et al., 2017; Jung, 2014; Wilhelms et al., 2017). Moreover, larger samples may also allow for subgroup analysis and the inclusion of heterogeneous groups, ages, and demographics into value-oriented studies (C. F. Lin et al., 2020; Rzepka, 2019). *Thirdly*, values remain challenging to evaluate due to their ambiguity and variability (J. Park & Han, 2018). Interviews that follow value-oriented approaches allow to probe for and analyze values in a structured fashion. Further, it provides a way to quantify and compare the results of multiple interviews. While the MEC approach has seen some application in IS research, I am not aware of studies that extended Jung's work by involving a wide audience outside of South Korea and exploring both positive and negative gains and values.

3.2.3 Methodology

In the following, I describe the research design and the procedure of the qualitative study, and the underlying method to investigate the real-world applicability of AI-based technologies to engage wide audiences in qualitative research. I compare a chatbot laddering interviewer (Ladderbot as introduced in Study I) with two implementations of the traditionally used PP laddering based on participants' behavior and perception during the interviews and the resulting insights. Additionally, I present findings from the hierarchical goal structure developed from content coding of the collected interview data across all three treatments.

3.2.3.1 Design and Procedure

I collected data in a between-subject design with three treatments: web-based paper-and-pencil laddering (PP), web-based visualized paper-and-pencil laddering (VPP) , and chatbot laddering with Ladderbot (LB) . Treatments are different from each other in terms of interaction (online survey vs. chatbot) and visualization of the interview history (off vs. on). The study was conducted as an online experiment. I chose an online experiment rather than inviting participants to participate in the study in an experimental lab, as an online experiment bears a closer resemblance to how I envision a chatbot laddering tool to be used in reality. Participants were randomly assigned to one of the three treatments. The interview for both the survey- and chatbot-based treatments mirrored the process of Jung (2014). I configured the laddering tools to follow an identical general structure by asking participants the following questions (Jung, 2014): *(1) What functions do you most frequently use on a smartphone? Please, write three examples*, *(2) Why do you use this function, and what do you obtain by using the function?*, *(3) Why is this reason (the last response) important?* After completing the laddering interview, I asked participants to fill out a questionnaire consisting of six dependent and various control variables. The experiment was conducted in German. I ran several pretests to ensure that the interview with the chatbot would work free of errors.

3.2.3.2 Participants

The experiment was conducted with a total of 381 participants in Germany, most of whom are students. One hundred eleven did not finish the experiment (dropout rate 29%), potentially due to the open nature of the inquiry, which required participants to invest more thought and time than, e.g., a quantitative survey. Of the 270 participants who completed the interview, 13 were removed as they failed two attention checks in the form of instructed-response items (Kung et al., 2018) included in the questionnaire (three participants) or completed the experiment twice as fast as the average of their treatment group (ten participants) (Meade & Craig, 2012). One participant was removed due to a technical error during the interview. A total of 256 participants were included in the analysis (M_{age} = 23.55, SD_{age} = 4.62; 42% female). Participants were heterogeneous with regards to highest completed education (high school 59.8%, Bachelor 28.1%, Master 8.2%, Ph.D. 2.7%, other 1.1%). Every participant reported using a smartphone daily. I incentivized participants by giving them the chance to participate in a lottery, where I raffled off 600€, with the highest prize being 50€, the lowest being 20€, and a total of 21 winners. Figure 3.4 outlines the data collection and analysis process of Study II.

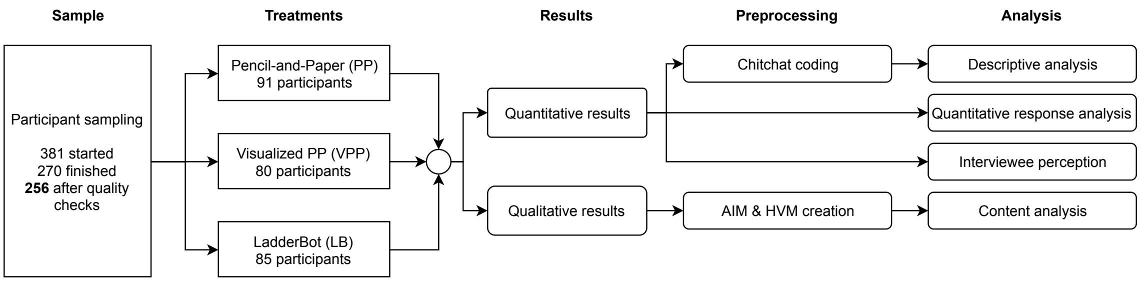


Figure 3.4: Data collection and analysis process.

3.2.3.3 Treatments

Online survey-based paper-and-pencil laddering (PP). The PP treatment replicates the well-established paper-and-pencil laddering in an online survey (Miles & Rowe, 2004). Participants are asked to name a frequently used function and subsequently answer the question "which is important to you, because. . . " a minimum of three times. Subsequently, participants can provide up to three more responses, if they wish to do so, for a possible total of six responses. While questionnaire-based laddering usually asks participants to provide a fixed number of responses, I chose to allow users to decide for themselves how many responses they wish to give (Gnewuch, Morana, & Maedche, 2017; Pieters, Bottschen, et al., 1998). This choice allows us to compare the willingness of participants to provide more than the mandatory number of responses.

Online survey-based visualized paper-and-pencil laddering (VPP). The second treatment is an identical copy of PP as far as the survey is concerned. Additionally, this treatment includes a visual representation of the interview history, which expands as the participants provide additional answers. The visual representation is identical to the one used in the LB treatment. I applied this treatment to control for an effect of the visual interview history on participant behavior and perception.

Chatbot-based using Ladderbot (LB). In the third treatment, participants conduct a laddering interview with Ladderbot. Ladderbot follows a hard laddering approach that requires users to complete a ladder for one attribute before changing to the next. During the interview, the chatbot switches between three dialogues, which control three segments of the interview: the elicitation of three attributes, a series of questions to elicit a ladder for each attribute, and a control dialogue to initiate each ladder and conclude the interview

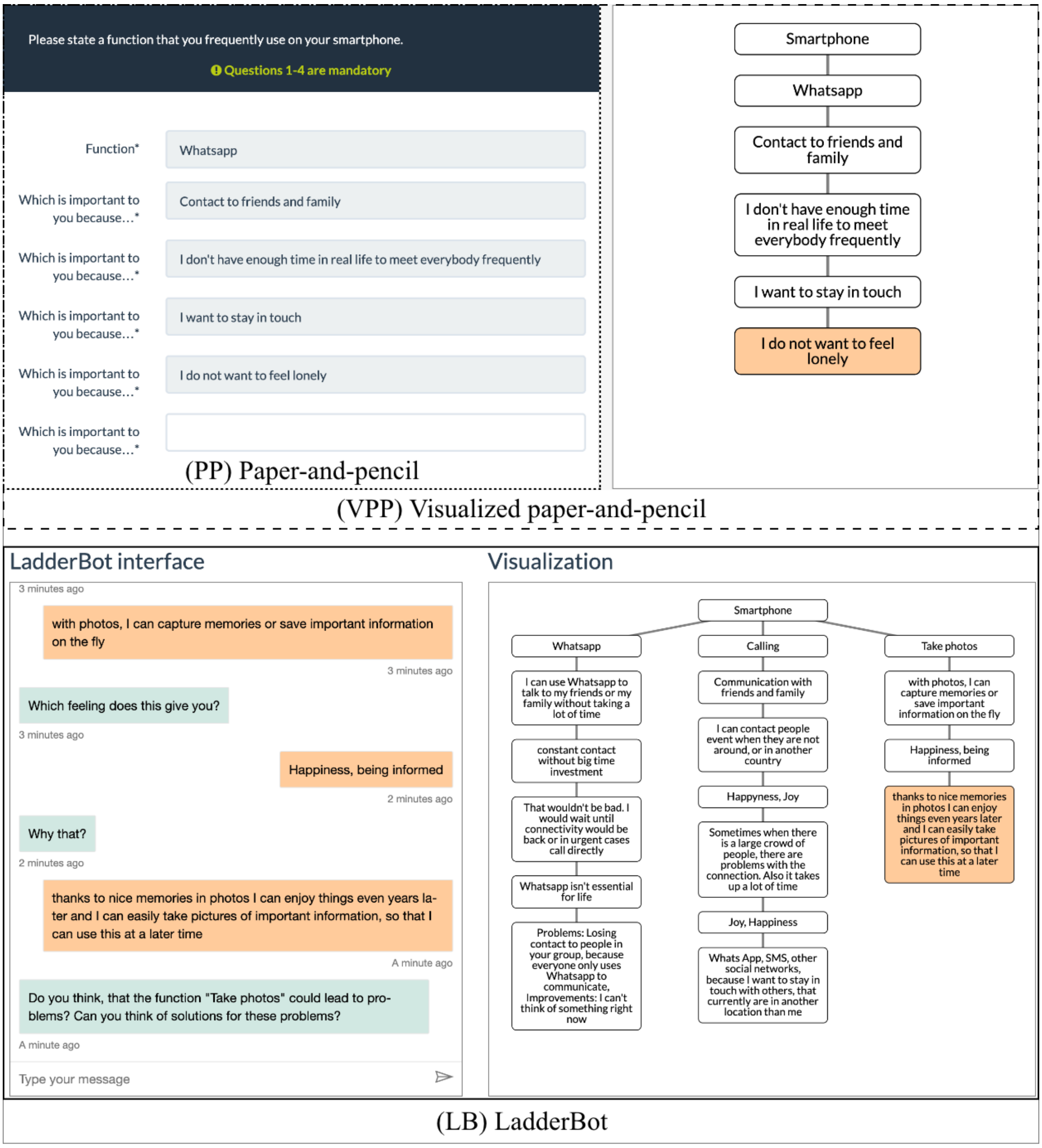


Figure 3.5: Interface of the three treatments: PP, VPP, LB. All elements translated from German.

process. Attribute elicitation, ladder introduction, and interview conclusion are identical for every participant. Refer to Section 3.1.2.2 for a detailed description of the interview structure and the guiding questions that Ladderbot used. Interviewees were required to answer at least three questions per attribute, which is identical to the mandatory questions in treatments PP & VPP. All treatments were integrated into a web survey instrument, Limesurvey, which combined the introduction to the experiment, the treatments, and the questionnaire. Figure 3.5 shows the user interfaces of the three treatments.

3.2.3.4 Quantitative Response Analysis

Several measures allow me to compare the three treatments. I calculate the average number of responses, the average number of words used per participant, and the average length of answers for each treatment. While the quantity and quality of interview data need to go hand in hand, these measures allow us to compare the volume of information that the evaluated methods collect on average via statistical means. While there is a limit to how much new information comes from more answers (Kaciak & Cullen, 2009), I perceive the quantity of collected data as a prime rating criterion for a data collection tool (Jeon et al., 2006). To identify "chitchat" – answers that are not relevant for the interview or intentional or unintentional errors (meaningless chunks of words/letters), I manually code each answer. The percentage of chitchat per treatment allows us to quantify one aspect of data quality. Finally, I investigate the number of times participants used the "stop" command in the LB treatment to move to the following attribute. I compare this percentage to the percentage of users in the PP and VPP treatments that decided to give more answers than mandatory. Further, I measure Understandability (5 items), Learnability (7 Items), Enjoyment (5 items), Efficiency (5 items), and Effectiveness (11 items) to compare the perceptions of interviewees about the different treatments. I adopted these measures from Coulin (2007), who applied them to evaluate a tool in a similar context (distributed requirements elicitation) (Coulin, 2007). All items were measured on a 7-point Likert scale ranging from "Strongly disagree" to "Strongly agree" (Vagias, 2006). Furthermore, I added questions related to satisfaction with the tool (7 items), Demographics (3 items), and Technology Experience (3 items, 5-point Likert scale) (Turel et al., 2011). I investigated the effects of my control variables on my results with a linear regression model. Subsequently, I applied Cronbach's alpha and inter-item correlation to assess the reliability of items and constructs ($\alpha > .7$ for all constructs. Some items were dropped due to weak correlations). I applied the Shapiro-Wilk test to test for a normal distribution of the constructs, which revealed that scores for all constructs are significantly different from a normal distribution ($p < .05$). Therefore, I apply non-parametric tests to analyze the questionnaire. I use Spearman correlation to evaluate correlations between constructs and Kruskal-Wallis tests to analyze the differences between the three treatments.

3.2.3.5 Coding

Table 3.5 summarizes my coding procedure of the interview data. In the first step, the first and the second coder categorized responses from a random sub-set of interviews (30 interviews each). Both coders classified the data using a mixed coding approach, consisting of deductive and inductive coding. The coders applied deductive coding by starting with the codebook established by Jung (2014) and classifying responses accordingly. If a response could not be classified using the existing codebook, the coders applied inductive coding to create new codes from the data. Coders created new codes descriptively, aiming to explain higher-level concepts. Differences between the two resulting codebooks from both coders were discussed together with a coding facilitator to form an elaborate set of codes and mitigate bias. In the result, seven functions/characteristics (mobile commerce, management of schedule and information, entertainment, communication, information search, social media, and basic device features) were extracted. Further, the coders extracted 24 consequences (of which eleven were negative gains) and twelve values. Based on the elaborated set, the first and the second coder classified the data from both the PP

	Participant	Action	Outcome
First step	First and second coder	Deductive and inductive coding (interviews subset – 30 each).	Subset of data coded Initial sets of codes
Second step	First and second coder and coding facilitator	Discussion of codebook(s) with coding facilitator. Development of elaborated set of codes.	Elaborated set of codes (7 attributes, 24 consequences, 12 values)
Third step	First, second, and third coder	Recoding using the elaborated set of codes and all interviews (PP and VPP by first and second coder, LB by third coder). Solving disagreements by discussions between coders and coding facilitator.	Final and complete version of data coded

Table 3.5: Coding procedure.

and VPP treatment again, while a third coder classified the data from the LB treatment. The final codebook is included in Appendix B.1.

3.2.3.6 Generating the Hierarchical Goal Structure

In laddering interview analysis, content coding is followed by the construction of an aggregate implication matrix (AIM) . This matrix represents the links between the concepts identified in the laddering interviews. The matrix aims not to represent individual ladders, but to produce an aggregate representation of the interview data, often referred to as implications (Miles & Rowe, 2004). Finally, implications in laddering are presented as an HVM, which is constructed from the AIM. The HVM presents the content and structure of the participants' knowledge regarding a topic in a graphical way. The next step in the analysis was the generation of the HVM for user values in using smartphones based on the data collected across all three treatments. I ordered the coded responses to form ladders of meanings. Therefore, the coding facilitator and I firstly assigned all codes to the three levels of abstractness: attributes, consequences, and values. The interviews conducted with Ladderbot required two additional levels, probes, and negative gains. Probes refer to the guiding techniques that Ladderbot used to gain deeper insights from interviews, i.e., exclusion and retrospective. Negative gains refer to the negative consequences that participants associated with an attribute. I summarized all relations between the elements of ladders in an AIM, which depicts the number of times that one code led to each other code in the responses (Miles & Rowe, 2004), as shown in Figure 3.6. My AIM represents the sum of direct and indirect relations, as both implication types should be used to construct the HVM (Miles & Rowe, 2004; Reynolds & Gutman, 1988). Direct relations are those in which one code leads directly to another, while for indirect relations, one content code leads to another with one or more other codes in between. Appendix B.2 shows the complete AIM.

At this point, the classification of codes to levels of abstractness is based upon subjective judgment. I utilize abstractness and centrality to evaluate these initial classifications and position codes in the HVM (Pieters, Baumgartner, et al., 1995). Abstractness and centrality are defined based on the in-degrees and out-degrees of a code. The out-degree of a particular element refers to the number of times an element serves as the start or origin

		<----Consequences												---->	<----Values											---->	In-	Out-			
	#	7	8	9	10	11	12	13	14	15	16	17	18	19	31	32	33	34	35	36	37	38	39	40	41	42	degrees	degrees	Centrality	Abstractness	#
A101 Mobile commerce	0	5	1	1	9			3	3			1			2	2			1	1	1	2			3		0	45	0,006	0,000	0
A102 Management of Schedule and Information	1	9	**54**	**16**	**43**	2	1	**24**	3		8	7	5		6	**33**	2	2		4	**12**	**13**	5	1	3	1	0	334	0,042	0,000	1
A103 Entertainement	2	**25**	**13**	3	**58**	3	**24**	**51**	**24**		**65**	11	1		7	**17**	8	6	9	**23**	**29**	**16**	8	5	3		0	800	0,102	0,000	2
A104 Communication	3	7	**57**	1	**59**	**28**	**29**	**81**	**23**	**28**		**31**	4	2	**25**	**15**	**99**	2	5	9	11	**16**	**27**	3	**12**	**28**	0	1460	0,186	0,000	3
A105 Information Search	4	**35**	**47**	4	6	**14**		**47**	**119**	4		**19**	3	2	**13**	**22**	6	3	**47**	6	**14**	**15**	**14**	1	7		0	804	0,102	0,000	4
A106 Social Media	5	2	7		8	**41**	1	**26**	**19**	**33**		**35**	2	1	1	3	**19**	1	5	5	4	4	9		1	11	0	409	0,052	0,000	5
A107 Basic device features	6	5	2	1	8	2		4	4		1	2			2	2		1		1	1		5				0	57	0,007	0,000	6
C201 Increased availability & flexibility	7		**14**	5	**13**	**17**	5	5	**13**	3	6	**14**			7	**13**	4		1	1	5	1	6		**12**		306	177	0,061	0,634	7
C202 Productive personal life	8	8		**12**	**18**	**19**	6	2	6	1	4	1	1		4	**41**	9	2	2	8	**18**	**19**	**12**	1	3	1	382	212	0,076	0,643	8
C203 Productive work life	9	1	2		1								1		2	11					4	7	1		1		92	32	0,016	0,742	9
C204 Simplification of physical tasks and positive substitution	10	**27**	**34**	8		**19**	11		2	2	**13**	**19**	5		2	**32**	**14**		8	11	2	**22**	8	1	6	2	444	304	0,095	0,594	10
C205 Enable & improve communication	11	**45**	**54**	8	**39**		**38**	2	**28**	**37**		**47**	6	1	**16**	**13**	**145**	1	6	9	**14**	2	5	1	9	**45**	533	609	0,145	0,467	11
C208 Sharing information and data	12	2	2	1	8	**14**			2	5	6	3	1		2	3	**16**		1		3	3				5	159	78	0,030	0,671	12
C209 No negative impact/indifference	13	1	1		2	2						6			1	1	1			1			3	1	1		516	30	0,069	0,945	13
C210 Extend general knowledge and inspiration	14	**24**	**29**	5	**26**	**24**	2	3		**12**	1	**28**	5		2	**15**	**16**	1	**65**	4	**12**	2	4		7	2	323	308	0,080	0,512	14
C211 Extend social knowledge	15		2		2	**18**	2	1	5			11	1			3	2		4	2	3	3			1	9	147	88	0,030	0,626	15
C212 Digital storage	16	1	3		8	1	**21**	2	8			**13**	1			**14**	4		1	**12**	8	1		3			120	110	0,029	0,522	16
C213 Feeling good and being entertained	17	8	**12**	1	6	**13**	4	3	**22**	**12**	9		1		2	**22**	11	3	9	**23**	**45**	11	1	2	4	4	410	238	0,082	0,633	17
C216 Improve health	18			2	2				2			2				6			1	2	1	4		1		2	40	25	0,008	0,615	18
C217 Source/Risk diversification	19				2											1											13	6	0,002	0,684	19
V401 Convenience	31				1	1			2	1		1				1	1				3				4		154	19	0,022	0,890	31
V402 Self-optimization	32		5	2	2	1	2		2				1				2			5	8	**12**			3		288	47	0,043	0,860	32
V403 Socialization	33				3	7	1	2	1	3		3	1		4	3			1	5	6	4		2	1	1	410	57	0,059	0,878	33
V404 Unobtrusiveness	34																						1		1	1	55	3	0,007	0,948	34
V405 Knowledge	35	1	1		1	1			3			1				2	2	1		1	2	5					190	21	0,027	0,900	35
V406 Hedonism	36				1			1			1	7				2	1				5	1		1			155	20	0,022	0,886	36
V407 Sense of comfort	37	1										2				4		2				1			1		237	11	0,032	0,956	37
V408 Satisfaction	38				2							2	1			4	2	2	1		5						227	19	0,031	0,923	38
V409 Safety and privacy	39	1	2		1	1						2			1			2	1	1		2		1	2		186	27	0,027	0,873	39
V410 (Mental) health	40																					1					32	1	0,004	0,970	40
V411 Autonomy	41	1			1											1			1	2	1	1					94	9	0,013	0,913	41
V413 Kinship	42	1	1		1	1		1		2		1					1					1					134	19	0,019	0,876	42

Figure 3.6: Abbreviated aggregate implication matrix.

(means) of a linkage with other elements (i.e., the row-sum of an element in the AIM). In contrast, the in-degree of an element refers to the number of times an element serves as the end of linkages with other elements (i.e., the column-sum of an element in the AIM) (Pieters, Baumgartner, et al., 1995). The abstractness of an element measures to which extend elements are predominantly means (at the beginning of ladders) or ends (at the end of ladders) in participants' perception. Specifically, it ranges from 0 (less abstract) to 1 (more abstract) (Miles & Rowe, 2004). Abstractness is calculated as the ratio of in-degree divided by in-degree plus out-degree of the element. (Pieters, Baumgartner, et al., 1995). The centrality of an element measures the extent to which an element is connected to all other elements in the AIM. Thus, centrality measures the importance of a concept in the means-end structure (Miles & Rowe, 2004). Its value ranges from 0 (less important) to 1 (more important). Centrality is calculated by dividing the total degree (in-degree plus out-degree) of a particular code by the sum of all active cells (no-zero cells) in the AIM. Across all treatments, the sum of all active cells for the current study was 7860. Next, I generated the HVM according to the information in the AIM.

I positioned the elements in the map according to their levels of abstractness and centrality and connected the elements according to their means-end relations. Since I cannot display all relations in the HVM without losing the map's usefulness and informativeness, I selected a cut-off level as the number of times that two codes have to be linked to be included in the HVM (Reynolds & Gutman, 1988). It is common practice to select a cut-off value that includes at least 70 percent of the implications derived from the raw data in the HVM (Reynolds & Gutman, 1988). I followed the method proposed by Bagozzi and Dabholkar (1994) and built Table 3.6 to arrive at a cut-off level of 12. This cut-off level represented 22.55% of the active cells and 75.93% of the active linkages. With the selected cut-off, the consequences *improve health* and *source/risk diversification*, and the values *unobtrusiveness* and *(mental) health* are excluded from the HVM because they do not have linkages with a value of 12 or higher. Finally, I compare the hierarchical goal structure created from the two survey-based treatments against the chatbot-based treatment. I build two additional AIMs, one for each technique, and subtract the linkages of the chatbot-based treatment from the linkages of the survey-based treatments. This subtraction allows us to identify and highlight the origin of specific linkages and compare the insights created by each of the two techniques. Appendix B.5 shows this subtracted AIM.

3.2.4 Results

I present two types of results: firstly, a descriptive analysis that compares the interaction of interviewees with the two survey-based approaches against the chatbot interviewer. Secondly, the generated hierarchical goal structures of user values of smartphones. Therein, I present the hierarchical map representing my results across all treatments and two HVMs that showcase the strengths and weaknesses of the two approaches to data collection.

3.2.4.1 Descriptive Analysis

Chitchat. I scanned the responses of participants for chitchat by scrolling through the list of answers and assigning "0" to answers that appear to be chitchat based on common sense

Cutoff level	Numbers of active cells in the implication matrix	Percentage of active cells at or above the cutoff level (%)	Number of active linkages in the implication matrix	Percentage of active linkages at or above the cutoff level (%)
10	206	25.53%	6218	79.11%
11	192	23.79%	6078	77.33%
12 (cutoff value)	**182**	**22.55%**	**5968**	**75.93%**
13	164	20.32%	5752	73.18%
14	151	18.71%	5583	71.03%

Table 3.6: Cutoff decision.

(e.g., exclude "sdfjs" as an answer, or single letters, e.g., "Z"). While this labeling is highly subjective, I only identified a tiny percentage of chitchat across treatments, calculated based on the overall number of responses (LB: 0.2%, PP/VPP: 0.27%). Across treatments, participants provided serious answers to the questions they were asked. The reason for the low percentage of chitchat, or the low percentage of participants that attempted to "play the system" or manipulate the experiment, might be that I explicitly communicated that the entry in the lottery is subject to a "complete and meaningful" participation. Hence, the results may differ when used with another incentivization strategy.

3.2.4.2 Quantitative Response Analysis

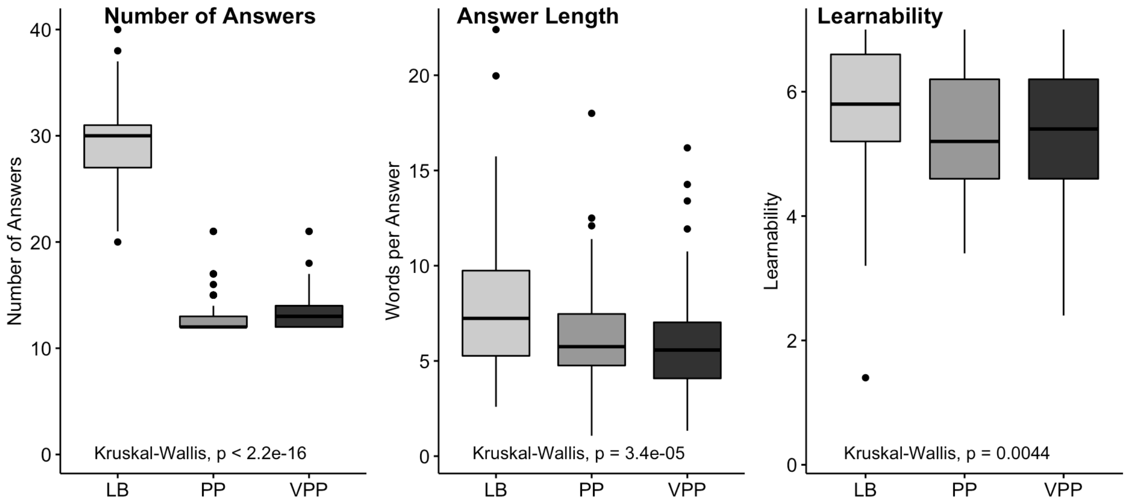
Control variables. I did not find any significant effects of the control variables gender, age, and education on the results presented in the following.

Response time. All treatments automatically recorded timestamps for all aspects of the experiment. I compare treatments based on the time taken for the laddering interview. Participants in the PP (M_{PP} = 7' 52", SD_{PP} = 4' 27") and the VPP (M_{VPP} = 7' 49", SD_{VPP} = 4' 26") treatment achieved a significantly faster time to complete the interview than the LB participants (M_{LB} = 16' 37", SD_{LB} = 6' 18"), $H(2) = 108.68$, $p < .001$. No participants with an extraordinary long interview time had to be removed, as the longest interview across treatments took 41' in the LB treatment (23' in PP and 26' in VPP, respectively). Refer to Figure 3.7 for a summary of the significant effects of Ladderbot, compared to the other treatments.

Average number of responses. I observe differences between treatments based on the number of responses, which I analyze using the non-parametric Kruskal-Wallis test, $H(2) = 180.3$, $p < .001$. Focused comparisons of the mean ranks between groups showed that the average number of responses was not significantly different between PP, $M_{PP} = 13$, $SD_{PP} = 1.93$ (1183 from 91 participants) and VPP, $M_{VPP} = 13.36$, $SD_{VPP} = 1.92$ (1069 from 80 participants) (difference = 14.15). However, the average number of responses for LB was significantly higher with $M_{LB} = 29.18$, $SD_{LB} = 3.69$ (2480 from 85 participants) compared to PP (difference = 134.49) and VPP (difference = 120.34). The critical difference (α= .05) for the comparison of LB – PP (VPP) was 26.74 (27.61), the critical difference (α= .05) for the comparison of PP and VPP was 27.17 (observed difference = 14.15). These results are in line with the significantly longer time that participants took for an interview with Ladderbot, as they were giving more answers.

Average number of words used. I applied the same methodology to evaluate the average number of words used. Similar to the numbers of responses, I observe a significant difference in average number of words between treatments, $H(2) = 136.24$, $p < .001$. Focused comparisons of the mean ranks between groups showed that the average number of words used was not significantly different between PP, $M_{PP} = 81.83$, $SD_{PP} = 38.48$ (7447 words from 91 participants) and VPP, $M_{VPP} = 78.97$, $SD_{VPP} = 43$ (6318 words from 80 participants) (difference = 7.34). However, the average number of words in the LB treatment was significantly higher, $M_{LB} = 229.58$, $SD_{LB} = 109.27$ (19514 from 85 participants) compared to PP (difference = 111.08) and VPP (difference = 118.42). The critical difference (α= .05) for the comparison of PP and VPP was 27.17, for the comparison of LB – PP (VPP), the critical difference (α= .05) was 26.74 (27.61). Additionally, I compare if the significant difference in avg. numbers of words can be explained by the higher avg. number of responses, or if participants simply give longer answers in the LB treatment. When comparing the avg. answer length between treatments, I find that there is a significant difference in avg. answer length between treatments, $H(2) = 20.6$, $p < .001$. Focused comparisons shows that participants in the LB treatment ($M_{LB} = 7.9$, $SD_{LB} = 3.61$) provide significantly longer answers than in PP ($M_{PP} = 6.24$, $SD_{PP} = 2.51$) or VPP ($M_{VPP} = 5.8$, $SD_{VPP} = 2.71$) respectively. While answers in the LB treatment are approximately 21% longer than answers in PP (difference = 36.3), and 26% longer than in VPP (difference = 50.4), compared to LB. The critical difference (α= .05) for PP (VPP) was 26.74 (27.61). Differences between PP and VPP are not significant (difference = 14.09), with a critical difference (α= .05) of 27.17. Figure 3.7 summarizes significant differences between the survey- and chatbot-based treatments.

"Stop" rate. The frequency of participants in the LB treatment using the "stop" command to switch to the following attribute is not directly comparable to when participants in the PP and VPP treatments gave more answers than they had to. However, it might serve as an indicator of the treatment in which participants were more likely to provide more information than required. Participants in the LB treatments used "stop" or its variations 73 times to switch to the following attribute. Consequently, participants only used the stop command in 28.6% of total ladders, based on a total of 255 ladders provided in total. The stop command was used by 45.9% (39/85) of the participants at least once. In comparison, 62.6% (57/91) of the PP participants and 46.3% (37/80) of the VPP participants provided only the mandatory number of answers. Three participants in PP (3.3%) and two in VPP (2.5%) answered all questions provided (21 in total, three attributes, and up to six why-questions). The difference between the number of ladders in which stop was used and the percentage of participants that used stop indicates that participants may have used the command when they felt that they had explained themselves in enough detail, rather than using it for lack of motivation.



	LadderBot		
	avg. improvement % based on mean		*Kruskal-Wallis*
Variable	PP	VPP	*p*
Response Time	111%	112%	< .001 ***
Number of Answers	124%	118%	< .001 ***
Number of Words	180%	191%	< .001 ***
Answer Length	27%	36%	< .001 ***
Learnability	8.3%	7.5%	< .01 **

Figure 3.7: Summary of significant differences between the three treatments.

3.2.4.3 Interviewee Perception Analysis

Using Spearman's correlation coefficient, I identified a significant correlation between my treatments and learnability, $r_S = 0.19$, $p < .01$, as well as enjoyment, $r_S = 0.11$, $p \sim .07$. Learnability is significantly correlated with number of responses, $r_S = 0.18$, $p < .01$ and word count, $r_S = 0.24$, $p < .001$. Efficiency is significantly correlated negatively with the treatments, $r_S = -0.12$, $p < .05$. Neither understandability nor effectiveness show significant correlations with my treatments. When comparing the three treatments for differences with regards to the identified significant correlations, I find that only learnability is significantly higher for LB, $H(2) = 10.84$, $p < .01$, while enjoyment is not significantly different between treatments, $H(2) = 3.44$, $p = 0.18$. While the results of the Kruskal-Wallis test did not show significant differences between the enjoyment of Ladderbot compared to the other treatments, Spearman's correlation coefficient suggests a positive effect of the LB treatment on enjoyment. Future research should evaluate the inclusion of various social cues into Ladderbot to increase enjoyment.

3.2.4.4 Content Analysis

The abbreviated hierarchical value map in Figure 3.8 reveals the most mentioned attributes of smartphones, what users try to pursue with them, and how consequences and values are related. The more abstract a code, the higher it is located in the map, and the more central a code, the more it is located in the center of the map (Jung, 2014). The complete HVM of positive gains is shown in Appendix B.3. The HVM contains the combined ladders of all three treatments. Users attempt to achieve several values by using smartphones, with the most frequently mentioned ones being *socialization*, *self-optimization*, *sense of comfort* and *satisfaction*. In terms of centrality of the codes, *enable & improve communication*, *simplification of physical tasks and positive substitution*, *feeling good and being entertained*, and *extend general knowledge and inspiration* have a predominant role in the means-end goal structure. These four elements are responsible for 18.56% of the outgoing and 21.76% of the ingoing linkages in the implication matrix. The most frequently used attributes or functions on smartphones include *communication* (e.g., messenger apps, calls, email), *entertainment* (e.g., camera, streaming, listening to music), and *information search* (news portals, navigation services, weather apps). The three attributes jointly correspond to 78.38% of outgoing linkages from functions in the implication matrix. In the following, I present the detailed findings.

Finding 1: Smartphones are predominantly communication devices to achieve social and utilitarian goals. The attribute *communication* is by far the most frequently mentioned function in the implication matrix (37.35% of all outgoing attribute linkages). Likewise, *enable & improve communication* is the most significant consequence, in that it has the most ingoing and outgoing (except for attributes) linkages in the AIM, and is linked to all of the most central goals. Participants use their smartphones as social devices to establish and nourish connections with peers, with the primary goal being *socialization*, followed by *kinship*. The ladder *communication* → *enable & improve communication* → *socialization* is the most relevant ladder in the HVM based on frequency. *Social media* and *information search* functions also serve as attributes towards the mean of improving communication. *Social media* provides a channel for staying in touch with family and friends and, in the same fashion as *information search*, enables users to stay informed about important life events. Participants seem to actively seek information to better communicate with others. As such, *extend general knowledge and inspiration* is a mean to *enable & improve communication*. In turn, the improved communication capabilities allow users to *extend social knowledge* and work towards the end of *socialization*. Further, the linkages between consequences demonstrate that smartphones have risen to the top spot in terms of communication equipment due to their capabilities for the *simplification of physical tasks and positive substitution* of direct physical communication. Phones help us stay in touch with a large peer group much faster and cheaper than ever before, resulting in an increased *availability & flexibility*. A positive effect of increased availability is that it allows users to achieve the goal of *autonomy*. In the Corona pandemic, for example, the increased availability and improved communication capabilities of phones (and other technology) allow employees to work from home and minimize social contacts. Consequently, users can achieve a *productive personal life*, thus helping them to work towards the goals of *self-optimization* and ultimately, *satisfaction*. At the same time, a *productive personal life* provides users with a *sense of comfort*, and a feeling of *safety and privacy*. Overall, participants use the various communication functions of their smartphones to communicate with friends, family, and coworkers, being able to increase their knowledge, flexibility, and *share information and data* as means towards the end of *socialization*, *self-optimization* and achieving a *sense of comfort*. Additionally, the social contacts that can be maintained

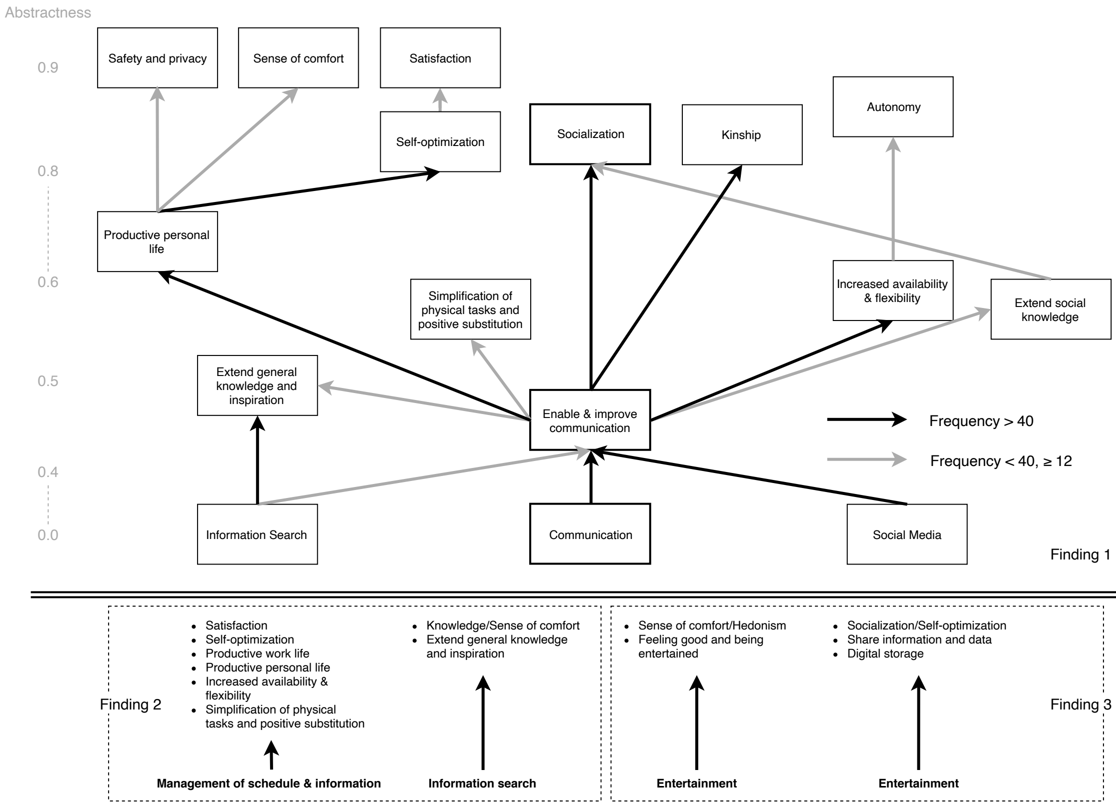


Figure 3.8: Hierarchical value map. Shortened for improved readability.

and nourished using smartphones help users achieve *kinship* by serving as a source of social validation.

Finding 2: Smartphone users seek intellectual and emotional self-optimization to achieve satisfaction. Second only to *socialization*, *self-optimization* is one of the most central values users attempt to achieve via their smartphones (13.32% of ingoing linkages for values, as compared to 18.96% for socialization). *Self-optimization* is not achieved predominantly with one attribute or consequence, such as *communication*, but is an end to many different means. The most significant ladder regarding outgoing linkages is *schedule & information* $\rightarrow$ *productive personal life* $\rightarrow$ *self-optimization.* Smartphones provide substitutes for various physical tools to support productivity, such as notepads, calendars, and to-do lists. My participants primarily include students – a user group that is well adjusted to using their smartphones to support their productivity. Not only do they use their phones to keep track of time, tasks, and appointments, but they also take photos of important slides and material during lectures or take notes during their studies. Via the now-ubiquitous app stores on every phone, users can access a myriad of apps supporting productivity improvement. These possibilities help users with the *simplification of physical tasks and positive substitution* and contribute to a *productive work life.* Overall, the capabilities of phones to simplify various tasks are a central means to achieve *self-optimization.* Besides providing access to functions that directly simplify schedule & information management tasks, users frequently use *information search* functions to *extend general knowledge and inspiration.* Phones do not only help users manage existing information but also allow them to acquire new knowledge.

Mediating factors that support the knowledge gathering capabilities of smartphones are the increased *availability & flexibility* (access to information at all times) and the *improved communication* (access to other sources of information). The *extended general knowledge and inspiration* in turn leads to a more *productive personal life* and allows users to pursue *self-optimization* and *satisfaction*. Further, as phones, due to their internet capabilities, allow access to information and knowledge just-in-time whenever necessary, they provide an essential means to allow users to achieve *knowledge*. While *self-optimization* is the most central goal that users pursue through a *productive personal life*, the simplification of tasks and extended knowledge, they pursue a *sense of comfort* as the most abstract goal (0.96). Once again, with my participants being students, for the most part, they seek a *sense of comfort* in dealing with the challenges of their job as students through productivity and flexible access to knowledge. The *schedule & information* as well as the *information search* functions, besides the *communication* functions, make smartphones one-for-all tools for both intellectual (productivity and knowledge) and emotional (sense of comfort and satisfaction) *self-optimization*.

Finding 3: Smartphones are feel-good and entertainment devices. Third to *communication*, the attribute *entertainment* is almost equal in terms of outgoing linkages to *information search* (800, 20.47% of outgoing linkages for attributes compared to information search with 804, 20.57%). Most significantly, *entertainment* functions help users to *feel good and be entertained*. The functions most commonly used by my participants were browsing the web for entertainment, using video-streaming, or playing games. Since the dawn of the smartphone era, the entertainment capabilities of phones have steadily increased, both due to hardware improvements, such as larger screens and faster processors, and new software services, such as Netflix and Spotify. While *enable & improve communication* also is an important mean towards *feeling good and being entertained*, a smartphone's entertainment functions are what users most frequently use to feel good and entertained. Ultimately, entertainment helps users to achieve a *sense of comfort*, which is a "state of ease and peaceful contentment" (Kolcaba & Kolcaba, 1991). While users achieve a *sense of comfort* via other means, too, the ladder of *entertainment* → *feeling good and being entertained* → *sense of comfort* is the most frequent ladder to this end. Additionally, feeling good links to *hedonism*, a state of pleasure and enjoyment (Pai & Arnott, 2013). Like how users achieve a *productive personal life* through productivity apps, the rise of the smartphone has also brought about dozens of entertainment-related apps. Users use these apps to simplify their access to or substitute "traditional" entertainment devices, such as the TV. A phone is often easier to use whenever entertainment is desired than a TV or dedicated gaming hardware. Thus, the entertainment functions of smartphones are easily accessible, in various situations, that can help users reach a context-dependent goal such as *sense of comfort* (e.g., watching a video), or *self-optimization* (e.g., listening to podcast on learning techniques). A second, highly relevant consequence of the entertainment attribute is *digital storage*. Users want to capture and store experiences and memories by using the camera function. This digital storage serves multiple purposes: being able to take pictures and store them without the need for an additional tool as a mean to experience *hedonism*; being able to *share information and data* with others towards the end of *socialization*;

and to augment the users own capabilities to memorize information towards the end of *self-optimization*. The camera function holds a special place in the hierarchical goal map, as it helps users achieve multiple functional, social, or emotional goals. To summarize, participants use entertainment functions on their smartphone to achieve a *sense of comfort* and *hedonism*, but also as a mean towards *socialization* and *self-optimization*, mediated by relying on a phone's *digital storage* capabilities.

Finding 4: Smartphones are not all about convenience. I understand the value *convenience* as anything that simplifies work and adds to one's ease (J. Park & Han, 2018). In the AIM, *convenience* is on the lower end regarding its centrality, with a mere 7.12% of ingoing linkages of all ingoing linkages for values. Participants achieve *convenience* through the means of a *simplification of physical tasks and positive substitution* and *enabled & improved communication*. Importantly, smartphone users are much more frequently striving for the ends of *socialization* (18.96%), *self-optimization* (13.32%), *a sense of comfort* (10.96%), and *satisfaction* (10.5%). On the other hand, *convenience* (7.12%) and *hedonism* (7.17%) are important ends to specific means, but are not the most sought after by smartphone users. To summarize, smartphone users value the capabilities of smartphones to simplify many aspects of their lives, such as communication, entertainment, and productivity functions. However, the participants use smartphones to achieve goals that are different from (just) the simplification of work or regular tasks. Moreover, they are striving for social contact and improving the own work and personal life, or more abstractly, being at ease and satisfied with themselves.

Up to this point, I looked at the information in the AIM that I collected across all three treatments, not differentiating between my two modes of data collection: survey-based and chatbot-based. However, the prompts for negative gains included in my chatbot interviewer provide insights into some of the shortcomings with frequently used functions and the concerns users face. Figure 3.9 shows the hierarchical value map of negative gains of smartphone use. The complete HVM of negative gains is shown in Appendix B.4.

Finding 5: Smartphone functions are commodities. When faced with unavailability or issues during the use of any of the four most prevalent attributes, *communication*, *entertainment*, *management of schedule & information* or *information search*, smartphone users make use of a *technology substitute, evasion or downgrade* (30.69% of ingoing linkages from prompts). On the other hand, reactions to downsides of a specific function are diverse, linking to each one of the negative gains above the cut-off value. While issues or unavailability causes users to have *(strong) negative feelings*, switching to alternative apps or technologies remains the prevalent reaction. Further, users appear to face *no negative impact / indifference* towards the consequence of switching or downgrading. However, the linkage between *technology substitution* and *no negative impact* is rather weak based on its frequency. When faced with issues with frequently used functions, users appear to be using or to have access to enough substitutes to circumvent service downtimes. This indicates that popular functions used on smartphones are mainly commodities.

Finding 6: Smartphones promote or force behavioral change. As users highlighted various negative gains as consequences of being prompted about *downsides of a functionality*,

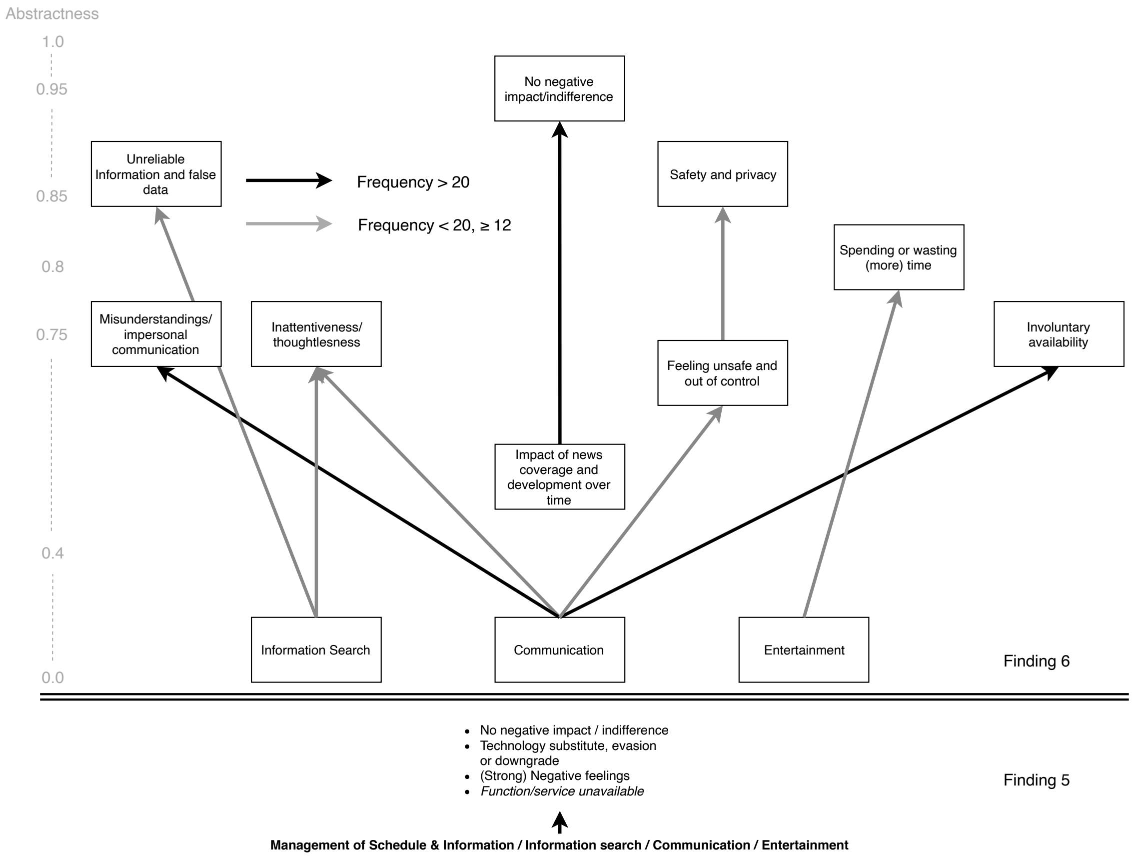


Figure 3.9: Hierarchical value map of negative gains. Shortened for improved readability.

understanding specific negative gains requires looking into the respective underlying attributes. In Finding 1, I introduced some of the positive gains of *increased availability and flexibility*. However, *involuntary availability* is one of the negative gains of *communication* functions for smartphone users. They commonly carry the device with them at all times, making them susceptible to unwanted distraction (Vaghefi et al., 2017). While easily staying in touch with social contacts is valuable to users, managing a magnitude of social contacts can require changes in communication behaviors, such as shorter and less focused communication (Nardi et al., 2000). In combination with the limited capabilities of the popular instant messaging services to convey non-verbal subtext in a conversation, this change can drive *misunderstandings and impersonal communication*. On the other hand, having the urge to continue a conversation in a messaging app, or just using the phone to check for new messages or news, can leave users with *inattentiveness and thoughtlessness* regarding events in the "physical world" (Dumitru et al., 2018; C. Wang & Lee, 2020). This negative gain is also linked to *information search*, as users might be tempted to disrupt a conversation by checking up on a conversation detail on their phones. Furthermore, having *entertainment* functions constantly available can hurt productivity, as it can seduce users to *spending or wasting (more) time*. I also identified a linkage between *communication* and *feeling unsafe and out of control*, which is a (negative) mean towards the end of *safety and privacy*. Various large and small scandals over the past years regarding data privacy in communication services and the illegitimate usage of personal information may have impacted the users' perception of the security of their private conversations (Kehr et al., 2015). However, when prompted on the *impact of news coverage and its development over time*, smartphone users largely did not perceive a negative impact. Finally, smartphone users face the negative gain of *unreliable information and false data* when searching for information.

3.2.4.5 Differences between Survey- and Chatbot-based Laddering

The quantitative response analysis showed a significant difference between the two survey-based and the chatbot-based treatments regarding the number of answers (121% more answers on average) and the answer length (31.5% longer answers on average). Consequently, I was interested in analyzing how the difference in answering behavior translated into the AIM and the HVM. Overall, while participants in both treatments produced similar ladders, both data collection types differ heavily regarding abstract codes and the end of ladders. Chatbot-based laddering produced much fewer ladders that end with a value or highly abstract code (sum of in-degrees for values in survey-based laddering: 1423, sum of in-degrees for values in chatbot-based laddering: 739). Similarly, values are also much less linked in chatbot-based laddering (sum of value out-degrees for surveys: 178; sum of value out-degrees for chatbot: 75). On the other hand, linkages between consequences and between attributes and consequences are much more similar between the two collection types (sum of in-degrees for consequences in survey-based laddering: 1562, sum of in-degrees for consequences in chatbot-based laddering: 1923). Besides the difference in interactivity between the two approaches, the chatbot-based approach went further than survey-based laddering in that it included prompting for negative gains. As such, participants spent part of their responses talking about their negative gains, resulting in Findings 5 and 6. I observe almost 80% more outgoing linkages from attributes in chatbot-based laddering than in survey-based laddering (2511 compared to 1398). Roughly 25% of these linkages connect to negative gains (620). Further, participants in the chatbot treatment derived *kinship* from the improved communication that smartphones provide them with, a linkage that did not surpass the cutoff-value in the survey treatments. Similarly, users in the chatbot treatment describe concerns that inhibit them from achieving a feeling of *safety and privacy.* Such concerns are fostered by the means the *simplification of physical tasks* and related to a *productive personal life.* Users are concerned that they lose control of their highly personalized or sensitive information, given the multitude of applications in use. With more digital communication tools being used both in private and at work, users feel easier to track and control. Finally, users voice concerns about the simplicity with which information about them can be shared due to smartphones without giving their consent (e.g., photos). Finally, participants interacting with the chatbot perceived *enable & improve communication* as a mean towards *feeling good and being entertained*, a linkage below the cut-off in the survey treatments. The chatbot interviewer prompted participants directly about their feelings regarding the improved communication capabilities. In the surveys, participants may not necessarily have thought about the feelings connected to improved communication directly, but instead focused on linkages to other means (e.g., *productive personal life*) or ends (e.g., *socialization*).

3.2.5 Discussion

In Study II, I explore the consequences and values that users pursue through smartphone use by applying the value-oriented perspective to data that I collected from a wide audience. Therefore, I apply and compare two approaches to large-scale data collection: online surveys and chatbots. My findings suggest that users try to achieve multiple interconnected goals and values through a smartphone, of which the most dominant are *socialization*, *self-optimization* and a *sense of comfort*. The study has theoretical, methodological, and practical implications, which I detail in the following section.

3.2.5.1 Implications for Research

This study has two objectives that deliver important theoretical and methodological implications: (1) to understand user values in smartphone usage today with European students, and (2) to use and compare state-of-the-art tool-based approaches to involve a wide audience in qualitative research. Therein, the study tackles several limitations of previous work. *Firstly*, I present results from applying laddering techniques that allow for conducting studies with large sample sizes – a well-known limitation of value-oriented research in IS. Further, I present results from interviews with European students, which can help broaden the understanding of values in smartphone use from the oftentimes Asia-based previous work. *Secondly*, I present a bottom-up view of smartphone acceptance and values in which I investigate both positive and negative gains. With this approach, I provide an alternative perspective on smartphone usage compared to common top-down approaches (e.g., focusing on addiction). *Thirdly*, smartphones and their integration into everyday life have changed significantly since 2014. This study helps to update the body of knowledge and demonstrate the evolution of user goals in smartphone usage.

User Values in Smartphone Use

Jung's initial study provided more vivid explanations of smartphone practices (Jung, 2014). For the interviewed South Korean students, utilitarian values of smartphone usage constituted intermediate goals towards achieving confidence in themselves. Overall, *confidence* was the most central value in the hierarchical map, hinting towards the qualities of smartphones as a user-empowering device. Further, Jung stressed the socioemotional characteristics of IT, outlined by the connection between social factors (socialization) and hedonic factors (amusement). Jung suggests that future research should examine these socioemotional and user-empowering characteristics not only with regards to smartphones but also for other IT environments. While I began my analysis process by adapting the codebook of Jung to my data, I quickly found that I had to discard *sense of confidence*, as defined by Jung, as not relevant for my AIM. My participants did not mention a feeling of superiority towards others, and feeling confident about one's abilities was only reflected as aspects of the values of *self-optimization* or *satisfaction*. These differences could be resulting from demographic differences between the two samples and affect my results compared to Jung's findings.

My hierarchical map shows that smartphones are predominantly a means to communicate. This finding is in line with recent related work, where texting, social networking services, and calls are most frequently used (J. Park & Han, 2018). Communication is crucial for participants to achieve *socialization* and *kinship*, through being valued by peers. As users strive for social value in smartphones, alongside a sense of comfort, my findings reiterate the importance of the socioemotional characteristic of IT to understand user behavior. The hierarchical map further demonstrates that convenience, while remaining an important value for users to achieve through the ability of phones to simplify and improve, is not the most significant value for users. Rather, my findings suggest that users primarily seek *socialization* and *self-optimization*, and secondarily *satisfaction* and a *sense of comfort*. Meanwhile, they might be willing to sacrifice some *convenience* (e.g., use paid-for rather than free apps) to better reach primary ends.

The self-optimization that users seek from smartphones has a professional and an emotional side as a means to achieve satisfaction. Using a phone to augment the own capabilities is an important concept I can discover in my results, with linkages to various concepts. As such, the concept of self-optimization may have implications for the related concept of self-efficacy, which indicates a situationally specific confidence to execute a task (Bandura, 1977). In contrast, self-optimization in this study, as the desire of a user to acquire or improve specific abilities or knowledge to gain an advantage, may describe somewhat of an antecedent to efficacy. In the sense of gaining an advantage, it is essential to understand what the reference point is in this comparison – the "old" self or a social comparison point, e.g., a friend or colleague. When elucidating a theoretical meaning of self-optimization, researchers could use self-efficacy as a mediator between self-optimization and a sense of satisfaction. Such studies may demonstrate the significance of self-optimization for common outcome measures and help better understand the connection between improving oneself, feeling confident, and its influence on satisfaction. Extending beyond the individual, demonstrating the significance of self-optimization, and investigating factors that affect the desire to self-optimize (e.g., social norms, work pressure, and personal aspiration) can help IS research understand more about negative gains of user-empowering IT. Insights from these research directions may inspire an update to the established types of embodiment in consumer technology (Hedman et al., 2019). We might see a type of user that utilizes technology as a natural extension of the own self to reach a new plane of human capabilities, the *augmentationalists*. While augmentationalists may look to increase comfort, capabilities, or confidence via technology, other user types (e.g., *conditionalists*) may be warier about wasting time as part of routine-based, time-in use (Bødker et al., 2014). Looking particularly at functions whose utility value lies in entertainment as a means to achieve a good feeling and a sense of comfort, bottom-up studies can help to better understand IT use. Specifically, studies that distinguish time-out use of entertainment and negative gains, such as addiction, can help us better understand subconscious decisions that impact how much augmentation users allow and desire in everyday use.

Further research along these lines can also allow us to better understand the impact that expected negative gains of technologies have on usage behavior. My study makes a contribution by showing the negative gains that smartphone users have, especially in communication functions. In particular, issues such as involuntary availability are a hot topic in light of the convergence of personal and workspaces as home offices become more prevalent (Dery et al., 2014). Given that many of the apps used on smartphones are at risk of becoming commodities, a growing awareness of shortcomings and risks of increasing augmentation could lead users to become more conscious of time-in use and strive for a conscious time-out use. Users may voluntarily make such behavioral adjustments for both productivity- and pleasure-related functions to circumvent negative gains, such as an always-on mentality, impersonal communication, or unreliable information. With technology getting closer to users (e.g., smartwatches, augmented and virtual realities), having a value-oriented, bottom-up view on how users interact with technologies that augment their capabilities and how new capabilities might forcefully change their behavior will be critical to user-centered research in ISD.

Strategies for Wide Audience Laddering Interviews

As I could not identify significant differences between the treatments PP and VPP, I consider them two instantiations of survey-based laddering. Comparing survey-based laddering against chatbot-laddering on the grounds of descriptive, quantitative response, perception, and content analysis creates mixed results. Overall, I value both approaches, as they come with individual strengths and weaknesses.

My survey-based treatments used a simple hard laddering structure without any added prompts to probe for negative gains. This structure resulted in easy-to-analyze ladders that commonly ended in values. Chatbot-laddering was similar regarding attributes and consequences, but ladders connected to values much less frequently. As participants in parts left the hard laddering structure that the survey followed and were prompted to talk about negative gains in a more semi-soft laddering style, ladders commonly did no end in values. The social aspects involved in conversing with a chatbot, following the computers-are-social-actors (CASA) paradigm, may have influenced the focus of participant's answers. Interviews ended primarily on social values, such as kinship, potentially due to the more interactive, social style of collecting answers. It would be interesting to understand better how interview modalities may nudge interviewees towards specific themes. While survey-based laddering helped us understand the primary values achieved by smartphone use, chatbot-laddering shed light on additional aspects that did not come up in the surveys. The notions of kinship and safety and privacy, for example, were mentioned only in the chatbot treatment with high enough frequency to appear in the HVM. Further, the chatbot outperformed the surveys with regards to participant engagement and higher learnability. I find that participants interacting with my chatbot provide twice as many answers, with individual answers being more than 20% longer than the treatments PP and VPP. Furthermore, participants show higher learnability, indicating that they had an easier time interacting with the chatbot than with survey-based laddering approaches. Despite participants committing more effort into chatbot interviews, this behavior does not come at the cost of a lower enjoyment. Finally, using an interview chatbot can be beneficial for eliciting more personal and detailed answers than manual interviews (Newman et al., 2002). This is especially beneficial for sensible interview contexts, such as addiction or abuse (Pompedda et al., 2017).

Researchers conducting laddering interviews with wide audiences will consider the complexity and costs of setting up possible tools to support their research. Overall, manual interviews are arguably the most complex to conduct for researchers, followed by chatbot interviews, with surveys being the most straightforward method. While surveys provide commendable results and help acquire insights with minimal setup and process costs involved, they may fall short of capturing intricate details that may be uncovered during manual or chatbot interviews. Therefore, I suggest a combination of manual interviews and chatbot or survey interviews. Manual interviews help researchers to get familiar with participants and get a feeling for the problem domain. Chatbot interviews are preferable to surveys for their positive effect on engagement and learnability and their effect on social desirability bias (Newman et al., 2002). Further, they may be set up to follow the same hard laddering structure that surveys do, circumventing the issues that this study had with semi-soft laddering. However, if a research team has no easy access to chatbots, surveys can provide commendable results.

3.2.5.2 Implications for Practice

This study offers insights for players in the IT sector, such as app developers, hardware providers, or communication companies. Specifically, my results can inform research and development of smartphones and apps focusing on user values and goals. Users do not use apps solely for their convenience but for their social or utilitarian value. Providing a strong offer focusing on extracting the worth of smartphones for self-optimization, socialization, or satisfaction (e.g., fitness tools, health tracking, learning) can help app providers to compete with free apps. Certainly, apps need to provide a well-designed user experience on top of their core offer. Overall, users switch apps easily if there are problems, making the differentiation from competing offers crucial. With the growing awareness of smartphone's negative gains, such as compromised privacy, impersonal communication, or time waste, mobile industry players may explore these user concerns to improve and distinguish their offerings. Therein, they may follow steps taken by hardware providers (e.g., Apple) towards providing users with functions to manage their consumption behavior and track security and privacy settings more easily. My results may hint at the shift from an attention-economy evolving around time-in use to a more conscious, attention-aware (time-out) use of apps and devices. Companies that adjust their offerings accordingly find themselves with a valuable differentiation from competitors. Furthermore, my results can inform marketing strategies on how to communicate the perks of smartphones to customers. Specifically, marketing campaigns should be developed around the notions of communication and socialization. Alternatively, promotions could center around possible productivity improvements related to professional or private contexts or highlight the devices' entertainment value, particularly the camera function.

My results have additional implications that extend beyond development and marketing. As productivity and self-improvement are essential for phone users, companies should (further) nourish strategies for guiding employees to use their devices proactively at work. While companies frequently provide employees with work smartphones, their integration into work practices varies. Some companies, e.g., from the IT-consultancy sector, provide employees with subscriptions to self-study portals and provide apps to perform organizational tasks (such as tracking work times or requesting refunds for travel expenses). Overall, organizational commitment can drive improving work efficiency through smartphones beyond private use. However, companies must also consider the negative gains associated with phones, particularly as the line between work and private life is blurring with the rise of remote work. While work phones are commonly used nowadays, many employees desire separate phones for work and private life. This is neither user- nor environmental-friendly – companies need to find new ways towards enabling employees to feel in charge of their availability and how they spend their time.

3.2.6 Conclusion

As smartphones have become an essential feature of human life and are here to stay, I set out to re-evaluate user values of smartphones. I believe that with a widely adopted technology such as the smartphone, a large number of user voices should guide my understanding of the phenomenon. Therefore, I demonstrate the application of modern approaches to data collection from a wide audience of users. The results revealed that while users associate diverse goals and values with smartphone use, I can identify primary ends that guide usage behavior. Further, the results highlight negative gains of the strong diffusion of smartphones into professional and private aspects of daily life. I believe that the study will inspire other researchers to broaden their scope to include a wide audience of users in value-oriented approaches.

3.3 Study 3: Cody - An AI-Based System to Semi-Automate Qualitative Coding

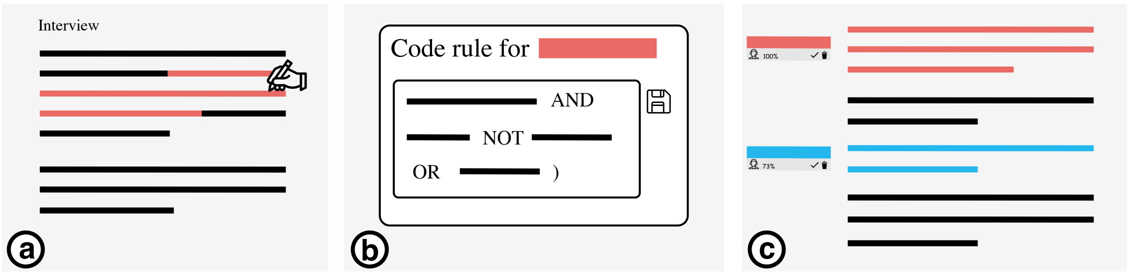


Figure 3.10: Cody used to extend qualitative coding to unseen data. (a) The user makes an annotation in a text document. (b) The user revises a rule suggestion to define the created code. (c) Cody searches text for other occurrences (red), and trains a supervised machine learning model to extend manual coding to seen and unseen data (blue).

3.3.1 Introduction

Qualitative research is valued not only in the HCI community to produce detailed descriptions and rounded understandings, allowing researchers to answer *what is?*-, *how?*-, and *why?*-questions (Ritchie & Lewis, 2003). It relies heavily on primary data in the form of unstructured text, transcribed from sources such as recordings from interviews or focus groups. The annotation of transcripts with descriptive or inferential labels referred to as *coding*, is an essential step for making sense of the text to drive the development of concepts or theory (N.-C. Chen, Kocielnik, et al., 2016). Within qualitative data analysis (QDA), coding is iterative. It goes from identifying initial categories in data during first-pass coding to assigning and revising labels to identify categories and themes. While qualitative researchers cherish good coding as a mix of science and art (Ritchie & Lewis, 2003), detailed and extensive texts make coding highly time-consuming and error-prone. Much of the process can be painstaking and repetitive (Xiao et al., 2020). This challenge is further aggravated with access to more massive datasets with new possibilities for scalable data collection (Rietz & Maedche, 2019; Tallyn et al., 2018), causing coding to lose reliability and become intractable (Abbasi, 2016; N.-C. Chen, Drouhard, et al., 2018).

QDAS aim to support researchers during qualitative coding and analysis with MAXQDA, Nvivo, Atlas.ti, Dedoose, WebQDA, and QDAMiner being commonly used (Freitas et al., 2018). Some of these systems incorporate ML to accelerate qualitative coding based on human annotations (De Almeida et al., 2019; Nvivo, 2020; Yimam, Biemann, Eckart de Castilho, et al., 2014). However, recent user studies demonstrated two critical shortcomings that impede the utility of available systems for enabling qualitative coding at scale (N.-C. Chen, Kocielnik, et al., 2016; Drouhard et al., 2017; Marathe & Toyama, 2018): (*i*) QDAS do not integrate ML as an interactive process that involves refining automated suggestions. The system mostly restricts the interaction between the user and the ML model to accepting and rejecting codes without insight into underlying coding rules. (*ii*) Therefore, code suggestions lack transparency, causing qualitative researchers the be reluctant to adopt ML-based support for qualitative coding.

Study III addresses these gaps by designing and evaluating a novel interactive AI-based ML system to support qualitative coding. Building on the recent work of the HCI and the IML communities, I present Cody, a user-facing system for semi-automating coding. I present the results of two evaluations: Firstly, a formative evaluation to understand *how qualitative researchers interact with and whether they would trust an IML system to support coding?* Secondly, a summative evaluation, investigating *how qualitative researchers use Cody compared to the commercial and well-established QDAS MAXQDA?*

My novel contributions include the following: I explain the design of the AI-based system Cody, which allows end-users to define and apply code rules (Figure 3.10b) while training a supervised ML model to extend coding to seen and unseen data (Figure 3.10c). Therein, I propose ideas for tackling challenges such as generating suggestions for code rules and cold start training of the ML model. Through interviews with qualitative researchers, after having used Cody for one week, I found that compared to MAXQDA, automated suggestions increased coding quality rather than coding speed. Further, while working with suggestions introduces an extra step to coding, this step is beneficial for researchers to get a better overview of the documents and to reduce the workload in the long run. Additionally, researchers desired explanations, particularly for ML-based suggestions, but rarely worked with them during the coding process. Finally, I discuss gains in intercoder reliability when using Cody; implications for designing suggestions to be *less* precise but *more engaging*; and meta-issues around automated suggestions for qualitative research.

3.3.2 Cody

Cody emphasizes an interactive AI-supported coding process. Users can specify their desired unit-of-analysis, add annotations and codes, define coding rules, react to suggestions, and access a rudimentary statistics page. Figure 3.11 shows the interface of Cody during the coding process. This section details the requirements for Cody to support the coding process successfully.

3.3.2.1 System Requirements

I defined six requirements to build an assistive tool for qualitative coding that pays attention to the HCI and AI challenges posed by qualitative data analysis (Wiedemann, 2013). The requirements are inspired by the excellent user-centered study presented by Marathe and Toyama (2018) and other related work (Rietz & Maedche, 2020). By satisfying the following requirements, I build a system that may act as a stepping-stone towards Wiedermann's vision for qualitative research: "*In combination with pattern-based approaches, powerful visualizations, and user-friendly browsers, [machine-learning algorithms] are capable to extend traditional qualitative research designs and open them up to large document collections*" (Wiedemann, 2013, p. 349).

- *R1 Unit-of-analysis.* The unit-of-analysis (UoA) defines the level at which annotations are made to the text (e.g., flexible or sentence-level). The system should allow users to set a UoA for a document to improve consistency between multiple coders (Crowston, Allen, et al., 2012; Marathe & Toyama, 2018).

- *R2 (Re)Define code rules.* Code rules can urge coders to combine keywords to form precise coding instructions (Ganji et al., 2018). Thereby, researchers might increase their understanding of the data (Grimmer & Stewart, 2013). During the coding process, coders encounter unexpected responses that affect previously defined code rules. As such, the system should enable coders to define and iteratively adjust code rules, applying the bounce technique (Paredes et al., 2017) (Figure 3.12d).

- *R3 Seamless training of ML model.* Qualitative researchers' primary goal is not to train an ML model but to identify meaningful instances in data (N.-C. Chen, Drouhard, et al., 2018). The system should require the user to be responsible for reviewing ML suggestions while hiding model and training complexity (Basit, 2003) (Figure 3.12f).

- *R4 Iterative suggestions based on manual annotations.* As researchers value coding parts of their data to familiarize themselves with the material while desiring recommendations to reduce repetitiveness, the system needs to incorporate manual annotations and update accordingly (Marathe & Toyama, 2018).

- *R5 Foster reflection.* In qualitative coding, imprecise codes become apparent as data is re-coded by a second coder, triggering an iterative code revision process (Richards, 2002). Code suggestions might act as a proxy for a second coder, as immature code rules help coders identify potential coding errors and enforce coding rigor (N.-C. Chen, Kocielnik, et al., 2016; Marathe & Toyama, 2018). The system needs to enable researchers to spot potential issues to reflect and iterate on coding rules (Figure 3.11c).

- *R6 Include explanations.* Suggestions need to be easily understandable to enable coders to predict how changes affect suggestions without requiring technical literacy (N.-C. Chen, Drouhard, et al., 2018; Cheng et al., 2019). Without understanding the source of suggestions, coders not trained in ML techniques might reject suggestions altogether, while novice coders might accept suggestions too easily. The system should explain suggestions by referencing code rules or highlighting relevant keywords, and providing a certainty factor (Figure 3.12e).

3.3.2.2 Coding Process with Cody

I developed Cody as a web-based system running on *Vue.js* (front end) and *Flask* (back end). Cody asks users to choose a UoA once a document is uploaded, which determines whether Cody automatically adjusts annotations to encompass an entire sentence (*R1*, Figure 3.11a). When applying a label to a selection, the user can use the label menu to review and adjust code rules by editing the rule in the text area (*R2*, Figure 3.12d).

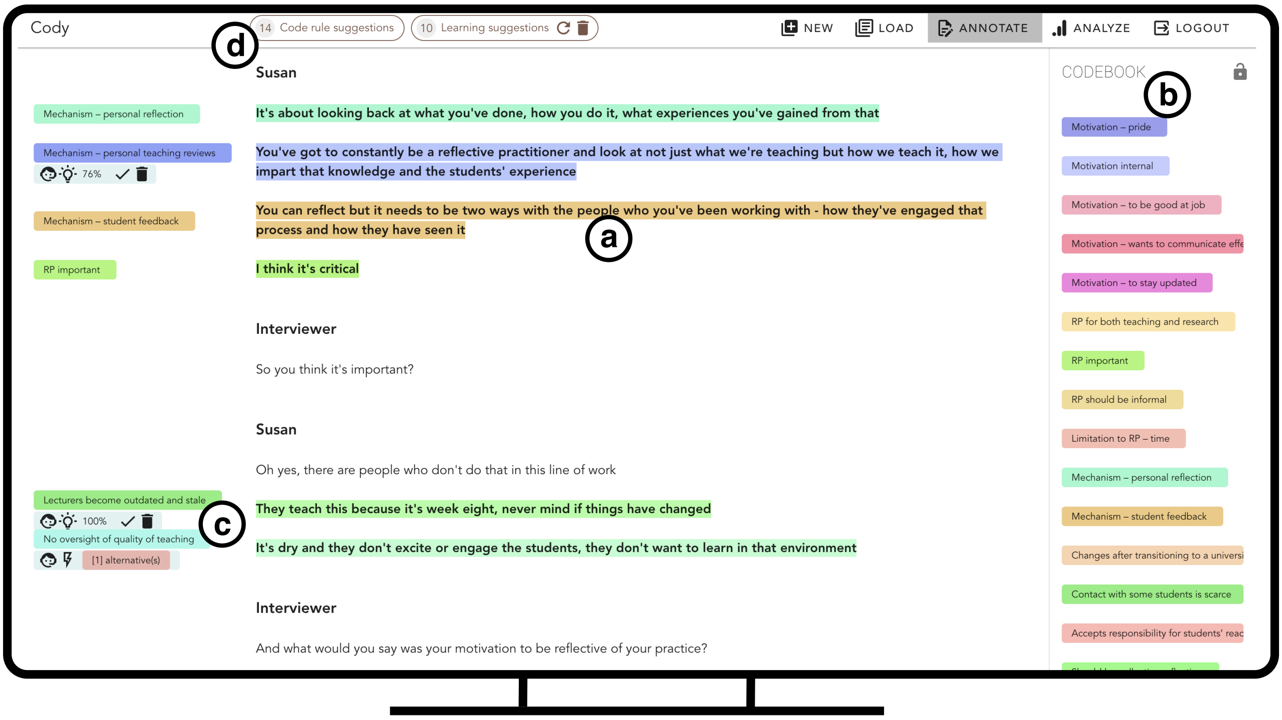


Figure 3.11: Final user interface of Cody. (a) main annotation view, (b) codebook sortable via drag-and-drop, (c) Code suggestion with confidence and accept/reject buttons. Below, Cody highlights multiple alternative suggestions for a section, (d) Number of rule- and ML-based suggestions

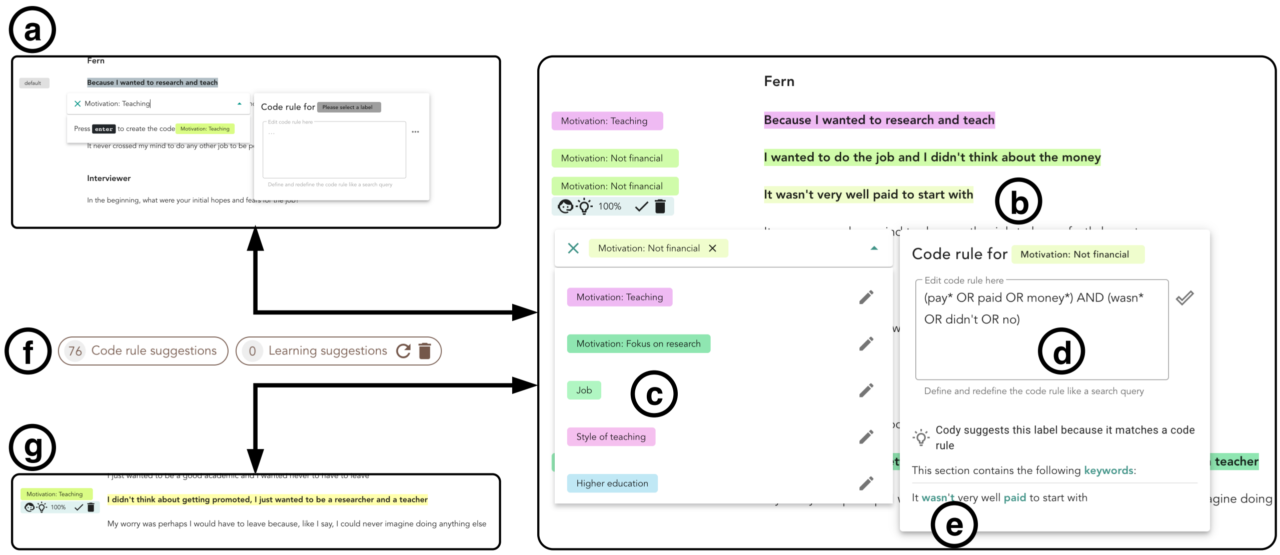


Figure 3.12: Coding workflow with Cody. Users make a new annotation and define a new code (a) which opens the code menu (b). Users may add codes to or delete codes from an annotation or edit a code (c). Cody suggests a possible code rule that users can edit (d). When clicking on suggestions to open the label menu, Cody shows explanations (e). Code rules are applied on saving to create suggestions and can be accepted/rejected by clicking the respective icons (g). The number of available suggestions is shown in the menu bar (f), where users can trigger ML model retraining (refresh icon) or delete all ML-based suggestions (trashcan icon).

Upon saving changes to rules, the new rule is applied to the entire document to create new suggestions. Users can review suggestions by clicking on either the label or the annotation, e.g., to revise conflicting code rules (*R2, R5*, Figure 3.11c) or to view explanations for suggestions (*R6*, Figure 3.12e). ML-based suggestions are updated automatically after ten manual changes to annotations (adding, editing, deleting) or whenever the user clicks the *refresh* button (*R3, R4*, Figure 3.12f).

3.3.2.3 Suggesting Labels with Code Rules

When a user creates a new code, the system generates an initial code rule suggestion. Therefore, the system compares the new code with the words of the respective annotation using *similarity scores* (SiS) and *Levenshtein distance*[7] (LD). I use *spaCy*, a Python library for NLP, to calculate SiS. Initially, I remove stopwords[8], spaces, and punctuations from the annotation. Depending on the text's language, the system then uses a pre-trained model in German or English. It compares the context-sensitive tensors of each word in the code with the lemmatized remaining words in the annotation to identify potential synonyms for codes that exceed an arbitrary cut-off value (similarity > 0.45). I use the LD to additionally include words in the rule that have a close enough match (relative LD > 0.3)[9] to the given code. Rule suggestions are lowercased, and no word can be contained twice. Initial code rule suggestions have the following form:

$$
\begin{aligned}
&rule \rightarrow \; lemmatized(LD\ 1) * \; AND\ lemmatized(LD\ n) * \\
&AND\ [\ lemmatized(SIS\ 1) * \; OR\ lemmatized(SIS\ n) * \ ]
\end{aligned}
$$

Whenever Cody generates a new rule, or when a user changes a rule, Cody applies it to the entire document upon saving (Figure 3.13). I use the Python library *whoosh* (Chaput, 2020) to search documents and identify occurrences (Marathe & Toyama, 2018). I structure every document in sections to make code suggestions. In a typical interview transcript, each sentence will form one section. When a rule changes, whoosh parses the code rule into a search query and applies it to the indexed document, returning the IDs of matching sections. Cody relies on section IDs to update (add & remove) annotation suggestions on the front end. Thus, the system makes suggestions on the sentence level. Currently, code rules will not automatically account for syntax or spelling errors in the underlying data (e.g., interview transcripts). Users may include wildcards in code rules which allow for *softer* matches to handle noise. Further, Cody highlights matching keywords for a suggestion in the label menu below the rule input text area. For rule-based suggestions, Cody highlights matched words in an excerpt from the current annotation (*R6*).

[7] The Levenshtein distance can informally be defined as the minimum number of single-character edits (insertions, deletions or substitutions) that are required to change one word into the other.

[8] Stopwords are words that occur with a high frequency independent of textual genre, e.g., 'the' in English (Marathe & Toyama, 2018).

[9] I determined cut-off values for similarity scores and Levenshtein distance through iterative testing of labels, annotations, and resulting rules suggestions. As such, the cut-off values are arbitrary, and other values will result in a different balance of words in the suggestions.

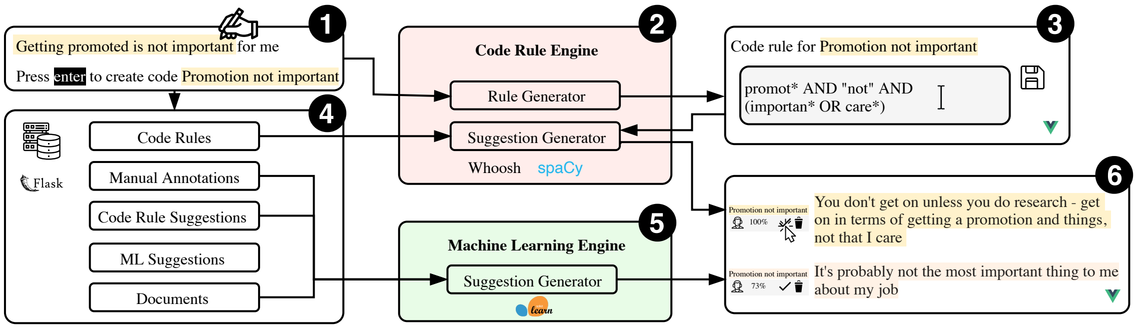


Figure 3.13: System architecture. (1) User makes an annotation, (2) code rule engine generates rule suggestion, (3) new rule is displayed for user review, (4) save triggers suggestion generator to search indexed document for occurrences, (4) and sends an update to suggestions in the database. (5) Machine learning engine retrains model and makes suggestions, (6) displayed for user review in the front end.

3.3.2.4 Suggesting Labels with Supervised ML

One crucial challenge to making code suggestions through supervised ML is the availability of labeled examples (cold start problem). Cody utilizes both manual annotations and rule-based suggestions to kick-start training the ML model (*R4*). As supervised ML algorithm, Cody trains a *logistic regression with stochastic gradient descent (SGD) learning*[10] to classify unseen data based on the available annotations (positive examples) using *scikit-learn* (Pedregosa et al., 2011) (Figure 3.13). I use the words in annotations as features for training while removing language-depended stopwords. For preprocessing, I used most of the default settings of the TfidfVectorizer[11] from scikit-learn to create a learnable matrix of TF-IDF[12] features. In coding, researchers usually work with more than two labels, making the classification of sections a *multiclass* problem. In the multiclass case, I deal with a low number of positives for each label and lack explicit negative examples (annotations indicating the absence of a label). Cody creates artificial negative examples to increase training data by treating unlabeled sections of text above the last manual annotation as negatives, assuming that the user makes annotations from top to bottom. Introducing artificial negatives (*greygoo* labels) also enables the algorithm to mark a section as "not relevant" if the predicted label is greygoo. Furthermore, I draw inspiration from the *S-EM algorithm for PU learning*[13] to create a threshold for inaccurate suggestions (Schrunner et al., 2020). I sample spies (S) from the labeled training data (L) through a test-training split, so that $|S| = 0.1 \times |L|$. After training the model with the available training data for all

[10] I compared various techniques for supervised learning according to precision, recall, f1-Score, and training and prediction time to select the most promising algorithm for my scenario. SGD fitting a logistic regression outperformed other algorithms (SVC, MNB, Random Forest, Logistic Regression, SGD with linear SVMs, Neural Network with LBFGS solver) with an f1-Score of 0.48. With hyperparameter tuning, I could achieve a label accuracy of .677 and an overall accuracy of .734, using a logarithmic loss function, balanced class weights, and the *elasticnet* penalty. While these values might seem unimpressive at first, the scores were achieved with a training set of 90 positive examples from eight different labels for predicting 721 unlabeled sections.

[11] Adjusted settings were sublinear_tf = True, min_df = 2, encoding = latin-1, ngram_range = (1,2).

[12] TF-IDF, short for 'term frequency – inverse document frequency', is a numerical statistic intended to reflect the importance of a word in a document or a collection of documents.

[13] S-EM: Spy expectation-maximization, PU: Learning from labeled and unlabeled examples.

codes (C), I predict labels for every spy (s). Cody will only display ML-based suggestions for codes (c) for that all spies were predicted correctly, thereby prioritizing precision over recall, i.e.,

$$c = \{\ c \in C : \forall s \in S : q(s|c) = s|c\ \}$$

with *q(s|c)* being the predicted spy-code combination for spy s and $s|c$ being the correct spy-code combination. When the model fails to correctly predict spies for each of the available codes, I deleted all existing ML suggestions.

My strategy of continuous real-time retraining of the ML model as the labeled data changes impacts the selection of an appropriate ML model, as low average training times are crucial. In my experiments, model training only took milliseconds, depending heavily on the amount of labeled training data. I expect frequent model retraining to be useful when the prediction model is less stable, which is the case with a low amount of training data – resulting in fast model retraining. As the amount of labeled data grows, the model should become more stable and would not need (re)training after every change.

For ML-based annotations, Cody displays counterfactual explanations in the form of indicative words for a suggestion to both help users understand the words of a sentence that the algorithm learned while potentially providing them with ideas for revising code rules (*R6*). The calculation of counterfactual explanations is comparable to the calculation of *Shapley Values*, which explain a prediction by highlighting the impact of individual features. Cody calculates the impact of a feature (each word of a sentence) by predicting a label while removing one word (or combinations of words) from a sentence (*R6*) (heuristic approach, c.f. Lindberg (2020)). Due to the computational costs of the pairwise comparison, Cody stops after iterating through all one- and two-word combinations.

3.3.3 Evaluation

During development, I conducted a formative evaluation to understand how researchers interact with my prototype(s), followed by a summative evaluation to compare the interaction with Cody against MAXQDA.

3.3.3.1 Formative

Formative evaluations aim at collecting information to improve an artifact (Ritchie & Lewis, 2003). Following the call-for-research for building and evaluating a user-facing interface (Marathe & Toyama, 2018), I firstly focused on evaluating how Cody's design, combining rule-based with ML-based suggestions, was perceived by qualitative researchers and determined necessary changes.

Method

I recruited participants following criterion-based sampling via a graduate university mailing list. Participants needed to be PhDs or PhD students with prior training in qualitative research who personally performed qualitative coding for at least one study in the last year.

Additionally, participants should have coding experience with a QDAS. Six PhD students agreed to participate, whom I invited for two subsequent iterations over two weeks.

I used contextual inquiry to guide the data collection (Bednar & Welch, 2009). Each session for both iterations consisted of three parts: (1) Introduction to Cody (5 mins), (2) in-situ evaluation with the think-aloud-method (25 mins), (3) Semi-structured interview on user experience (30 mins). I provided participants with a task description to follow while sharing their thoughts, ideas, and problems following the think-aloud-method (Fan et al., 2020). In the task description, I asked participants to perform three tasks: (1) Load their document into Cody. Participants gave me access to data from own projects, which I converted to a file type that Cody could process. (2) Switch to the coding view, and (3) Perform qualitative coding on the document by recreating the coding process applied when initially analyzing the data. While participants used Cody to code their dataset, I took notes while observing their progress on a second screen. Each session concluded with a semi-structured interview, during which I asked participants for the features they most liked and disliked; their perception of code rules; their perception of interface and coding efficiency; trust in suggestions; differences to their usual coding process and perceived usefulness; and their willingness to use Cody to automate coding partially. Appendix C.1 shows the interview guide for the semi-structured interviews.

I transcribed the audio recordings of each session. I conducted inductive coding on both transcripts and field notes, followed by discussions with a second researcher to iteratively refine emergent themes. I summarized findings on a per-participant level by comparing observations and aggregated findings to identify required and future improvements. My goal was to understand user's work practices with Cody, to improve the user-facing interface. I use pseudonyms for anonymity and present slightly edited quotes for readability.

Findings: First Iteration

I started with a prototype running locally on a laptop. While already having the final artifact's functionality, this prototype of Cody aimed to minimize the actions users would have to take to code a document. Code rules were saved automatically and applied with every change. Cody would retrain the ML model whenever users added or edited an annotation or when a code rule was applied. Due to the relatively small number of labeled data available for model training, the processing time for retraining was in the range of milliseconds. Further, the Cody prototype did not indicate how many suggestions it has created so far.

Participants could use Cody with their data and coding scheme, if only for a short period. Tom, who commonly works with grounded theory, found Cody useful for initial coding as part of open coding: "*I think it would help me with a certain number of interviews to be faster with initial coding. I always have to identify [security requirements from qualitative interviews with experts], that takes time but has only limited benefit.*" Participants found rules particularly relevant for studies with many similar interviews, where they can learn from an initial sample and use rules to reduce repetitiveness. Lana explains: "*I've roughly 81 interview pieces – it became very boring and repetitive. Because they are only short* *statements, no in-depth interviews [...], but until then, I learned enough to be able to define rules for the remaining pieces.*" Interestingly, participants felt responsible for incorrect suggestions, having defined the underlying rule themselves: "*it misused customer service, but because I made a mistake*" (Cora). Further, I did not know how participants would think about the quality of suggestions for code rules. The quality did not matter much, as participants required suggestions for rules primarily as examples to learn about the rules' syntax: "*not every researcher is familiar with code rules, that's why it's important that this tool suggests rules and also shows how they should work. Otherwise I think this wouldn't be used*" (Cora).

The first prototype iteration convinced us that automated suggestions are perceived as beneficial when applied correctly. However, participants reported that they desired more control over the generation of suggestions, a better way to accept/reject them, and to see the number of generated suggestions. I adjusted the prototype accordingly and deployed it to a server to enable a remote evaluation.

Findings: Second Iteration

The second prototype was accessible on the web. Compared to iteration one, I changed the interface to be more intuitive at the cost of requiring more user actions. As such, users now had to save code rules manually, triggering their application. Cody retrains the ML model once every ten changes to annotations rather than after every change. I made this change to reduce the frequency with which I confront users with new suggestions. Further, users can manually request model retraining and the deletion of all ML suggestions. The menu bar now shows the number of existing rule and ML suggestions. Users can accept or reject individual suggestions directly via button-click. I added user profiles to allow for multiple users working with Cody simultaneously.

Overall, participants perceived the second prototype as helpful primarily to structure documents better and faster. Josh explains: "*what you can do much better with this tool than with MAXQDA or other tools is to deal with a topic explicitly. I could go back now and look at everything related to customers, and then I could look at everything related to platforms and so on. I don't have that in the other [tools], I would work through the document linearly, jumping back and forth between topic blocks. And that's why this can improve the coding because I can focus much more.*" Eric thought Cody to help more by reducing workload rather than improving coding quality: "*of course, there would be fewer errors, but it would not directly improve the quality. I would expect myself to work correctly; it would rather make it easier for me.*"

However, participants also had concerns about using Cody: One, Seth was afraid of "*missing certain things*" mainly when using AND operators in rules. Second, Eric had prejudices towards ML and ignored ML suggestions, feeling that they "*cannot work with that little amount data.*" However, he would feel better once he had labeled "*three to four documents,*" which would also help him to define code rules: "*to create good code rules, not only do I need coding experience, but I also need to know the text.*" Adding to this, Sven said: "*I think it makes a lot of sense if you let theory guide you and what you want to find in an*

Table 3.7: Summary of participant characteristics and statistics. Participants are pseudonymized. I use 'Disc' for discipline, 'Meth' for methodology, 'STS' for socio-technical studies, 'HCI' for human-computer interaction, 'IS' for information systems, 'GT' for grounded theory, 'MQ' for MAXQDA. For statistics, I use 'Ann' for annotations, 'Acc' for accepted suggestions, 'R chg' for number of changes to rules, 'ML ref' for number of manual ML refreshs, time in hh:mm, 'Pre' for precision, 'Rec' for recall, and 'GG' for including greygoo examples for training. Precision and recall are taken from the final model retraining.

Formative												
I1	*Disc*	*Meth*	*QDAS*	***I2***	*Disc*	*Meth*	*QDAS*					
Cora	IS	Iterativ	Miro	Eric	IS	Deductiv	MQ					
Lana	HCI	Inductiv	MQ	Josh	HCI	Iterativ	MQ					
Tom	STS	GT	Miro	Seth	HCI	Iterativ	MQ					
Summative												
Name	*Tool*	*Codes*	*Ann*	*Acc*	*R chg*	*ML ref*	*time*	*Pre (GG)*	*Rec (GG)*	*Pre*	*Rec*	
Ella	Cody	40	207	16	50	16	05:06	0.76	0.78	0.20	0.13	
Ena	Cody	37	383	139	31	23	08:26	0.61	0.56	0.58	0.36	
Kelly	Cody	52	119	3	51	9	04:56	0.83	0.81	0.00	0.00	
May	Cody	27	85	2	9	8	03:10	0.92	0.89	0.08	0.17	
Nas	Cody	36	173	48	20	10	06:47	0.82	0.81	0.50	0.38	
Paul	MQ	42	162	-	-	-	08:00	-	-	-	-	
Sana	MQ	40	114	-	-	-	05:30	-	-	-	-	
Stev	Cody	36	126	7	5	11	03:55	0.79	0.77	0.31	0.15	
Tabi	MQ	62	135	-	-	-	05:00	-	-	-	-	
Vic	MQ	23	101	-	-	-	05:15	-	-	-	-	
Zoe	MQ	27	152	-	-	-	03:30	-	-	-	-	

interview. If I use in-vivo coding, then code rules are of no use to me. But if I want to have some kind of structure, and want to break something down, then it makes sense." Participants felt that the usefulness of code rules lies in giving structure and that rules are best defined once they had familiarized themselves with the text. Eventually, automated suggestions would help to "*perceive the text as a whole*" (Josh), as it requires researchers to also re-read individual sections to review suggestions.

To summarize, participants perceived the automated suggestions of the second prototype to be most helpful for "*getting an overview faster*," (Eric) "*having a speed advantage*," (Seth) and building the codebook "*better, more stringent*" (Josh). Despite these benefits, Seth also noted that it would be a "*higher initial effort*," leading to coding "*becoming much easier.*" However, the interaction with the prototype was too short for participants to observe these effects for themselves. Josh explains: "*I can't judge this conclusively, you would have to do it with 20, 30, 40 codes to be able to say that.*"

3.3.3.2 Summative

A summative evaluation of an intervention or artifact is concerned with its impact on the effectiveness and resulting outcomes (Ritchie & Lewis, 2003). As such, I evaluated Cody's effectiveness compared to MAXQDA, one of the most well-known QDAS (Freitas et al., 2018). For the summative evaluation, I used the second version of Cody (see Figures 3.11 and 3.14a).

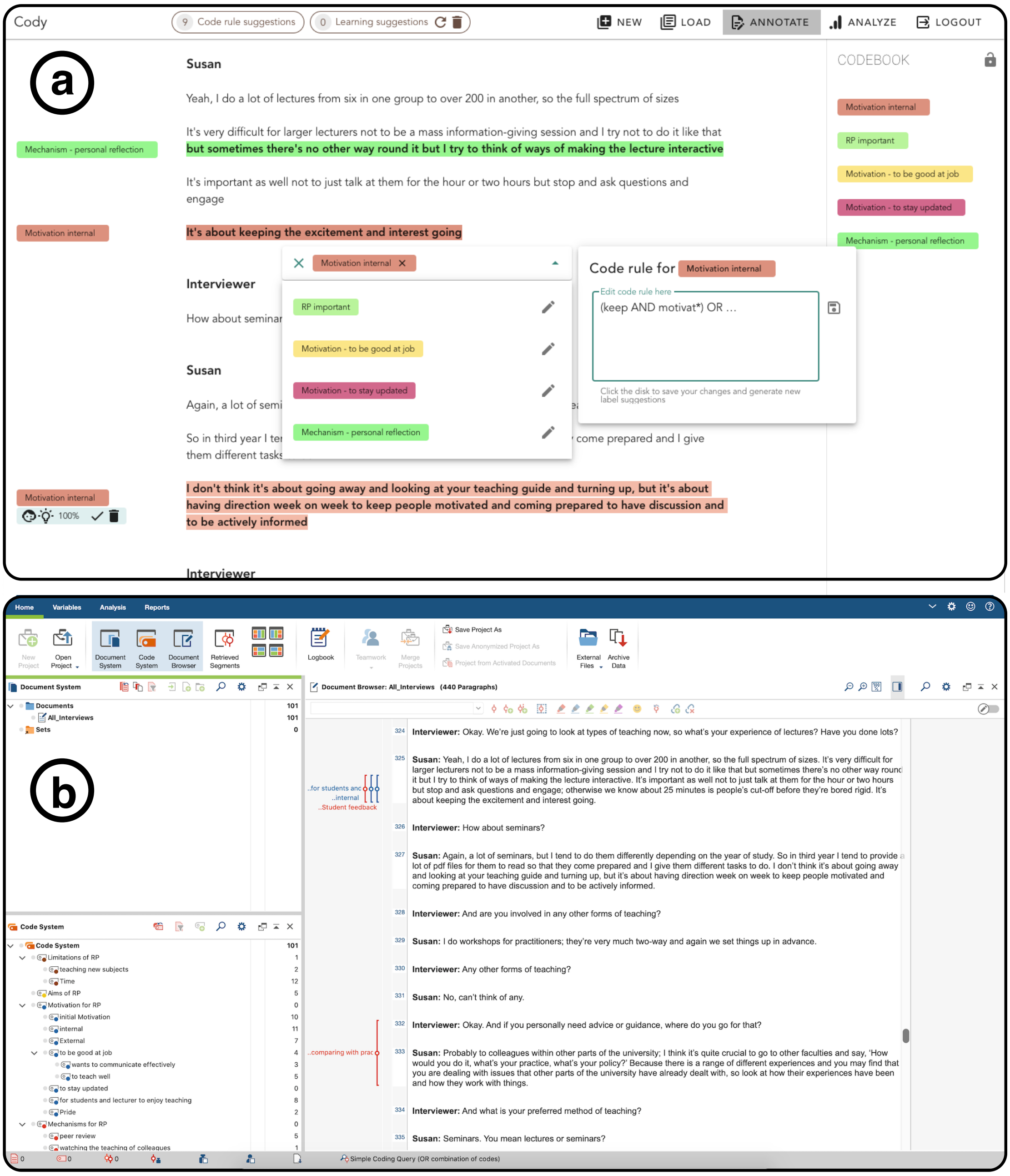


Figure 3.14: Screenshots of the user interface of (a) Cody and (b) MAXQDA. Both screens demonstrate what participants in their respective treatment saw during the summative evaluation.

Method

I invited participants from a pool of 3.500 university students using criterion-based sampling (Ritchie & Lewis, 2003): (I) Bachelor's degree, (II) performed at least one qualitative study, (III) experience with qualitative coding for at least one qualitative study and (IV) excellent English skills. I selected these criteria to ensure that participants are experienced in qualitative analysis. Eleven people ultimately agreed to participate. Table 3.7 presents a summary of participant characteristics both for the formative and summative evaluation as well as statistics of participants' interactions with their respective coding tools. I tasked participants with coding a dataset over one week in a between-subject design: one group using MAXQDA, the other Cody. Figure 3.14 shows screenshots of the interface of (a) Cody and (b) MAXQDA. I used a public dataset of interview transcripts on reflective practice in higher education (Harding, 2015). By evaluating Cody with a public dataset, I want to enable other researchers to evaluate future tools against the same dataset, as coding depends heavily on the underlying data. Furthermore, the dataset comes with a student guide for participants on how to code, steps to follow, and a complete codebook. Through the student guide, participants can evaluate the transcripts with a concrete goal: to *identify feelings about reflective practice and how it was put into practice* (Harding, 2015). Thus, I evaluated coding assistance with first-pass coding with a pre-developed codebook, as suggested by Marathe and Toyama (2018). However, participants were free to add new labels should they need to. At the beginning of the week, I invited participants to a 1-hour online workshop to introduce them to the task using the student guide, including a 15 minutes introduction to their respective QDAS. I conducted individual 30 minutes long semi-structured interviews with all participants after they finished the task. During the interview sessions, I asked participants about their coding experience with the QDAS compared to tools they are familiar with; their perception and usefulness of automated suggestions; explanations and effect on trust; and if they would use tools that semi-automate coding. Appendix C.2 and Appendix C.3 show the two interview guides for the semi-structured interviews. None of my participants in any study had prior experience with rule-based coding of qualitative data. I compensated participants with €90 for their time and expertise.

I transcribed the audio recordings of all interviews. I conducted inductive coding on the transcripts, followed by iterative discussions with a second researcher to refine emergent themes. While I could collect usage data from Cody, for MAXQDA, I partly rely on self-reported data from participants, such as the duration of coding. From participants' MAXQDA project files, I extracted the number of annotations made and the labels participants used. For Cody, I measured various parts of the interaction, such as the time taken to code, the number of manual or automated annotations, and how often code rules were adjusted. Based on the coded documents, I calculate Krippendorff's Alpha as a measure of intercoder reliability for both treatments (Krippendorff, 2004). The calculation of Krippendorff's Alpha required some preprocessing: I corrected spelling mistakes in codes and differences in the usage of symbols (- and –), which impact the calculation. For MAXQDA data, I transformed the data to match the export structure from Cody, to use the same calculation. I once again use pseudonyms for anonymity and present slightly edited quotes for readability.

I detail two types of findings: (1) *Impact of Automated Suggestions on Coding* highlights how rule- and ML-based suggestions influenced participants' coding. (2) *Implications for Designing AI-based Coding Support* presents three recommendations for automated QDA assistants.

Findings: Impact of Automated Suggestions on Coding

Code rules increase coding quality. An imprecise rule, when applied to an interview, creates multiple wrong suggestions. While participants needed some time to understand how to define rules at an appropriate scope, the process of iterating rules engaged them to think about their coding. Ella explains: "*it helped in the sense that I thought about: 'what does it have to contain to fit?'.*" Further, users tend to work with many overlapping labels. More precise definitions help to reduce overlap: "*as the codebook grows, I'm not even sure which code matches which text correctly. There are overlaps, that's why it's difficult if you haven't defined the codes correctly [...] I think it helps a lot to structure it much, much better from the beginning using exactly these keywords as search criteria.*" (Ena). Overall, participants reported having a better understanding of the coding scheme. As May puts it: "*we commonly work with definitions, but you don't see, it's mostly concepts, but not what words are relevant. Using [Cody], we have it clear and systematized.*" I was interested in seeing if the alleged understanding of the coding scheme translated to increased intercoder reliability (ICR), and calculated Krippendorff's Alpha. I selected Krippendorff's Alpha as a measure for ICR due to its applicability with six individual coders. In their insightful discussion of the value of calculating ICR, McDonald et al. (2019) argue that ICR can be a helpful measure when applying a codebook to data. For MAXQDA, five unique coders with an average of 132 annotations/coder had an Alpha of 0.085. For Cody, six unique coders with an average of 182 annotations/coder had an Alpha of 0.332. Also, rules are useful for understanding the work of other coders, mainly when code definitions are not discussed: "*It will be easier for third parties to understand. What was done, which rules were used to code the document* (Sana)."

However, the characteristics of the data and the aim of the analytical process determine the usefulness of code rules. The more structured the data, the easier it is to define rules that result in precise suggestions. Particularly with data from (semi)-structured interviews, rules can be fine-tuned to code specific sections of interviews (e.g., age and demographics) or responses to questions reoccurring across interviews (e.g., why did you decide to enter higher education?). Ella states: "*it depends on the questions and how standardized the whole thing is done. I could imagine if you have a lot of yes-no questions, it can help quite well.*" Luckily, interviews with a structure that suffices for rule creation also tend to be repetitive and time-consuming with little analytical reward. With interviews where meaning is hidden in context, code rules fail to provide useful suggestions as they discard dynamic semantics. Ella said: "*I revised [rules], often [...] if you think to general, you suddenly have 120 suggestions, then I changed it and had one. It's hard to balance, the answers can be the same but still so different, that the rule fails to find it.*" Further, code rules work best with an established codebook, e.g., when applying deductive coding. Lana states: "*If I don't have a codebook that I want to apply, I just try to see what is in [the document], without defining rules. But I think it makes a lot of sense if theory guides your coding and you want to find something from theory in an interview.*"

Despite the drawbacks of rules in dealing with context to make precise suggestions, participants also found rules to help structure data. Thereby, rules enable the scanning of documents for particular topics of interest. As Stev puts it: "*[Cody] definitely is a good support in the sense that, for example, I want to code everything related to motivation, then it takes work off my shoulders. Normally I would do this by hand using Ctrl+F and the mark relevant sections. This helps me not to overlook things.*"

To summarize, participants enjoyed working with code rules and used them not only to generate suggestions but also to re-think their coding. While they were not convinced that they could appropriately formulate rules for every type of code or data, they valued the feature for structuring interviews and increasing their understanding, especially for unfamiliar data. Participants using Cody had higher intercoder reliability compared to participants using MAXQDA.

ML suggestions should prioritize precision over recall. Cody's design purposefully hid the complexity of ML suggestions from the user. While some participants could barely tell whether they worked with ML suggestions, they valued not having to deal with rejecting multiple unhelpful suggestions. As such, systems should prioritize precision over recall when training ML models. Zoe explains: "*if I can only accept one of many suggestions, then it's a waste of time, because I have to check every time [. . .] So I'd rather have [suggestions] less often and more precise.*"

The low number of positive examples for each label is particularly challenging for model training, reinforcing the notion that a system should be careful not to distract the user with premature ML-based suggestions. Despite the low number of positive examples, Kelly had a positive experience with ML-based suggestions: "*those suggestions, that appeared without me changing [a code rule], this was something I didn't have before. And for some sections, where it made sense, it really reduced your workload.*" Further, participants were not distracted by having to reject wrong suggestions, given that wrong suggestions are not perceived as prevalent. "*A few times it really helped, but often I had to delete suggestions. Yes, I think it was ok. It's useful that the possibility exists at all*", Nas said.

Thus, ML-based suggestions are a double-edged sword. While they help to not miss exciting phenomena in the data, they lack quality when the number of positive examples is limited and require strict thresholds. In combination with code rules, ML suggestions are useful to extend suggestions to some of the *false negatives of rules*, supporting users in improving rules by highlighting instances that existing rules are missing. Hence, ML suggestions can support users if they focus on precision over recall, providing limited support while minimizing distractions. The coders' desire to work through their entire dataset additionally reduces the risk of missing relevant sections due to a low recall.

Checking suggestions is a beneficial extra step. Earlier user inquiries reported that researchers fear that automation would be adding one more step to coding, having to check not only what code the researcher would use but also what the computer said (Marathe & Toyama, 2018). All six participants working with Cody confirmed that while the coding process with Cody did not require them to change their general process, it took time to (re)define rules, and navigate the document, to accept and reject suggestions. Two participants quickly discarded checking seen data for new suggestions for a comprehensive check-up once they finished coding: "*towards the end, I didn't bother because I noticed that new [suggestions] would pop up anytime anyways. But especially in the beginning, I searched for them*" (Nas), "*maybe what was different than if I had done it with another software is that at the end I searched the whole interview for suggestions and either accepted or deleted them*" (Stev). An assistive system should make it easy for users to review suggestions, particularly those added to seen data. Ella and Nas suggest assisting users with reviewing new suggestions, thus reducing the disruption of the coding process. In Ella's words: "*When there are suggestions, I want to be able to go there and return to the position where I left.*" Further, reviewing suggestions for seen data had participants re-examine manual annotations and sometimes revealed sections that had been overlooked. Overall, on average, participants took similarly long to code the data between treatments (5:22 h with Cody to 5:20 h with MAXQDA). While I cannot draw conclusions regarding coding time due to the lack of internal validity, participants were convinced that using code rules can accelerate their coding process. However, they said that the number of interviews was too low to make appropriate use of rule-based suggestions.

Thus, reviewing automated suggestions, when provided not only for unseen but also for seen data, introduces an additional step to coding. While participants desired support on the interface level to review suggestions quickly, they did not perceive Cody's suggestions to negatively impact the coding procedure. On the contrary, Stev and Ena said that they used suggestions to double-check codes in a second-pass and get a better overview of the data as a whole: "*[...] you were brought to look more often, and without this help, you would have overlooked one or the other thing especially in the first run, you would have had to go through more often*" (Stev). Ena voiced the following when asked whether the automated suggestions helped: "*Yes, definitely. In the beginning, it was quite time-consuming to create all of them and to think about it. But it was cool when I had a page where five or six [annotations] were suggested, and I just had to read through and check 'do they fit, yes, no' [...] I really had the feeling that the work was easier.*"

Findings: Implications for Designing AI-based Coding Support

Provide suggestions at an appropriate level of detail. Especially participants using MAXQDA imagined suggestions not at a one-code level of detail visible in the text but as assistance to reduce the choice of codes for an annotation. Tabi explains: "*It would be nice if I had some suggestions [...] Maybe so that I only have to choose between five codes, so I don't have to look through all 30 codes when I make a selection in the text. Like three to five options.*" Further, Paul suggests to only highlight interesting sections without making code suggestions, highlighting potential sections of interest: "*the algorithm says, 'something could* *be here,' but you have to think for yourself if you want to do something with it, it would enhance you own process.*" Participants using Cody, on the other hand, showed little interest in simple highlights instead of suggestions. However, they were interested in multi-label suggestions. Kelly explains: "*you might lose the overview and accept [the suggestion] if only one code is suggested. But when you have several, then you can think about it again – which one fits best*?" There were two reasons for this preference. First, having three codes suggested strengthened users' confidence that the algorithm had considered all options. While the algorithm considered all choices for any decision, Sana felt that the algorithm might have missed something: "*With only one code suggested, you think 'has it really seen everything?' And with three, I would know that there is a higher probability that it selected the ones that fit.*" Second, participants felt at risk of accepting suggestions too quickly, particularly when being tired. Having multi-label suggestions requires users to make an active choice. Eventually, participants felt that this choice would help them trust the algorithm more. Sana explains: "*it remains transparent. Even when you have selected one out of three, you may still be able to see these three later. If you take your time to look at it again and see 'ah there it suggested these three, looking at it again, it still makes sense for me.*'"

To summarize, participants welcomed the idea of having suggestions not only provide one but three to five potential codes, increasing the involvement in decisions at the cost of additional work. It is primarily essential that a human is the last instance for reviewing suggestions, not allowing the system to "*auto-code* (Paul)."

Explanations are desired but get ignored. When asked about trust in and transparency of automated suggestions, participants using MAXQDA regarded explanations as elemental to understanding suggestions and working with an assistive system. While participants using Cody partly voice requiring explanations, they pay no attention to the explanations provided by Cody: "*There was something, but I probably didn't look at it very closely*" (Nas), "*generally, if they [suggestions] make sense, they make sense [. . .] I don't know if it's important that I see or don't see the specific rule*" (May), "*I verify that for myself and think about whether it can make sense*" (Ella). Primarily, participants are convinced by helpful suggestions. Sana explains: "*I would check it myself a few times in the beginning, and when I realize that it suggests the right thing, I would not doubt that in the future. I don't know if it needs a direct explanation.*" Hence, explanations should be provided, particularly on user request, but the perceived quality of suggestions decides the user's trust. Tabi explains that reading explanations is a trade-off, requiring time that could otherwise be used for coding. In Tabi's words: "*it would be nice, but takes time. The more explanations you have to read, the longer the process will take*". Eventually, the initial impressions are crucial for users' decision to adopt automated suggestions or ignore them (or turn them off). Further, users saw little value in the confidence scores I showed, saying that "*it would not strengthen my trust [. . .] having no idea how it was calculated*" (Sana).

Automation should encourage and support experimentation. Despite all users of Cody describing using code rules as "*new*" (Ella), "*exciting*" (Kelly), and "*interesting*" (Vic), they rarely started the task by trying to learn how to use them. Only Stev began coding by "*figuring out how to add a code, how to rename it, how do these rules look like, so I wrote an example with an asterisk to see if it automatically highlight the next line, which had such a keyword in it.*" Most participants took some time to figure out how to write code rules in a granularity that worked for their coding. Kelly explains: "*In the beginning, I may have formulated code rules a bit imprecisely, and it came up with suggestions which didn't fit at all. Then I always had to adapt by trial and error. But if you did it a couple of times, then it worked, then you learned how to formulate them in a way that gets you the results you want. And then [the suggestions] helped, because that's when you got suggestions that really fit.*" Participants did not actively look for more information or familiarize themselves with the tool before starting the task. Instead, they wanted to familiarize themselves with the functions and possibilities as they go. Ella explains: "*it's a learning-by-doing kind of process. The general introduction was enough. The rest you have to work out by yourself.*" None of the participants coded the entire dataset in one go, thus valuing on-demand introductions to certain features of a tool: "*I want to be able to say: 'Hey, now I want an introduction to the function.' Instead of being overwhelmed on my first use, why can't the tool remind me like 'Hey, how about trying the automation now?'*" (May).

To summarize, participants follow a learning-by-doing approach in working with code rules. An assistive tool should encourage experimentation and provide guidance or on-demand assistance while ensuring that users can test without fear. "*I would adjust rules and would work with it because I see the benefit. [...] What is important is that I know that no other labels disappear, that I lose nothing,*" Tom urged.

3.3.4 Discussion

3.3.4.1 Working with Automated Suggestions

With Study III, I pursue the goal of designing, building, and evaluating a user-facing system that integrates both prevalent strategies for (semi)-automating coding: code rules (Collins et al., 2019; Crowston, Allen, et al., 2012; Crowston, Liu, et al., 2010; Grimmer & Stewart, 2013; Marathe & Toyama, 2018) and (supervised) machine learning (Abrami et al., 2019; Klie et al., 2018; McCracken et al., 2014; Tietz et al., 2016; Yan et al., 2014; Yimam, Biemann, Majnaric, et al., 2015). Prior work on code rules has focused on evaluating rules defined by experts against gold standard datasets (Crowston, Allen, et al., 2012; Crowston, Liu, et al., 2010; Marathe & Toyama, 2018), while Cody focuses on enabling and supporting end-users in defining and reworking rules during coding. Through the formative evaluation with qualitative researchers, I identified the importance of rule suggestions to educate and encourage users to work with rules. While I drew some inspiration for automatically creating rule suggestions from the literature on text mining (Nakatoh et al., 2016), information extraction (Soderland, 1999), and classification (Takahashi et al., 2005), prior work at large did not focus on creating rules that are easy for users to read and edit. From my summative evaluation, I learned that while users had to change the suggested rules, as I intended them to, they valued the support and did not refrain from working with rules. Further, the final rules that users created were quite heterogeneous, some creating short (*Limitations to RP – time: time* AND [limit* OR less OR hard*]*) and some creating complex rules (*Mechanism – watching the teaching of colleagues: teaching* AND [colleagues OR others] AND ["learn* from" OR people* OR technique*]*). I also saw examples of generic rules, which could only be used to navigate a document rather than provide accurate suggestions (*Motivation – to be good at job: good* AND job**). While none of my participants were experienced with rule-based coding of qualitative data, it would be interesting to evaluate the impact of such experience on the interaction with code rules. Better initial results might create a positive reinforcement loop, reducing barriers for engaging with rule-based suggestions while fostering a positive perception of the tool. Overall, users were able to define rules that helped them to structure and, to some extend, speed up certain parts of the coding process. Thus, Study III extends prior work by demonstrating how users interact with code rules as coding support. With my work, I deliver new design implications for systems that integrate code rules and rule suggestions.

Regarding ML suggestions, I had to work around the cold start problem. Previous work required a minimum of 100 positive examples for each code (Yan et al., 2014), while participants in my evaluation, on average, only created 133 (MAXQDA) or 182 (Cody) positive examples overall. My participant Kelly reported the most interaction with ML suggestions[14], while others barely noticed them. I believe that the barriers I set for Cody to providing ML suggestions, namely defining cut-off values for prediction confidence and requiring labels to be predicted correctly for all test instances, helped filter out many wrong suggestions. In the summative evaluation, Cody trained the first ML model after participants made ten annotations and triggered model retraining after every ten subsequent changes. Further, artificial negatives allowed the model to determine a section to be neutral and refrain from making a suggestion. Participants perceived suggestions based on code rules as more helpful than ML suggestions. The strict quality criteria resulted in users interacting with only a low number of ML suggestions due to the number of positive examples necessary for the algorithm to make appropriate suggestions. My results and Cody's ability to extend coding more frequently to sections that do not match a code rule could be improved by harnessing strategies for tuning the ML model during usage. For example, Cody could allow the user to adjust cut-off value(s) for rule-based and ML suggestions. Overall, I expect ML suggestions to assist coders with improving code rules by identifying *false negatives* – sections that are not yet covered by a rule despite belonging to the underlying label. Enabling users to define perfect rules would eliminate the need for ML suggestions altogether. However, this might not be feasible given the costs involved in and practicality of defining ideal rules for certain qualitative research methods and data structure (Crowston, Allen, et al., 2012).

I calculated Krippendorff's Alpha to evaluate the coding consistency between my users, both for MAXQDA (0.085) and for Cody (0.332). As for the interpretation of an Alpha of 0.33, Kirppendorff suggests discounting conclusions from coding with an Alpha < 0.67

[14] For Kelly, the metrics of the last retraining of the model were: (Precision) 0.82, (Recall) 0.81, (F1-Score) 0.81, when including artificial 'greygoo' negative examples. Without them, metrics were: (Precision) 0.50, (Recall) 0.38, (F1-Score) 0.42. For training, 144 positive examples and 751 artificial negative examples were used. This training/prediction cycle resulted in 13 new suggestions for four labels that exceeded the cut-off.

(Krippendorff, 2004). Depending on the type of qualitative research, an Alpha of 0.33 can indicate that researchers/coders should discuss and improve the codebook in use. In the context of my study, using Cody resulted in an increased Alpha compared to MAXQDA despite including an additional coder in the calculation. While my experiment setup does not allow us to determine the cause of the difference in Krippendorff's Alpha, the result may provide a quantitative indication that supports my qualitative findings. I believe the difference to have two causes. One, as participants engaged with code rules and ML suggestions, they spend more time reflecting on their coding and going back and forth in the document to review suggestions, potentially also revising previous annotations. Two, Cody makes suggestions at the sentence level, which might have influenced the unit of analysis that participants used for annotations. While with MAXQDA, participants applied codes at various units (individual words – multiple paragraphs), participants using Cody quite frequently applied their codes on the sentence level, too. Thus, the way a system provides suggestions may influence how users code.

3.3.4.2 Researcher Agency and Reporting

While automated suggestions may serve as proxies for the second coder, they can impact researchers' agency. Especially participants with MAXQDA stated concerns about whether automated suggestions could impact coding quality, as coders would be tempted to accept suggestions to reduce their workload. As Cody's users told us that they rarely interacted with explanations, they are at risk of not realizing when a decision by the algorithm bases on incorrect or shallow assumptions (e.g., *higher* being an indicative word for the code *higher education*). However, participants felt responsible for the quality of their coding, and it was vital for them to get results that they can reliably use for subsequent analysis. One path to reduce the risk of carelessly accepting suggestions is to reduce the precision of suggestions by either: One, suggest not one but multiple labels and have the coder pick the most appropriate one. However, this approach would increase the time it takes to review suggestions. Two, suggest labels only when an annotation is made, rather than preemptively annotating sections in the text (e.g., in the context of semantic annotations, see Tietz et al. (2016)).

Regarding trust and agency, it also needs to be discussed where calculations are performed, be it for applying rules to documents or training an ML model on data. Qualitative data may contain sensitive information, and researchers might not always anonymize their data before coding. Thus, the user of an assistive system must have control over where data is processed and stored and can ideally run the system on their device or environment. Finally, researchers will only use systems for their projects that are accepted by their respective communities. Participants told us that they would not risk their work being rejected due to reviewers not being familiar with a new QDAS, particularly when authors would have to explain the tool's suggestion algorithm. While researchers would have to take responsibility for the suggestions they accept during coding, I believe that defining code rules can increase transparency in qualitative research projects, both for co-coders, reviewers, and other researchers. While code rules may not communicate all information that determines the application of a code, they can serve as an indication towards coding and allow, to some extent, the replication of results.

3.3.5 Conclusion

Inspired by previous work concerning AI-based qualitative coding, I set out to understand how real users interact with automated suggestions during coding. I designed and developed Cody, an interactive AI-based system supporting researchers with rule- and ML-based suggestions. I worked with qualitative researchers to iterate my designs, finding that given the proper assistance and interface, end-users would (re)define rules, convinced that it would help to improve their understanding, build stringent codebooks, and accelerate their coding. Based on my findings, I conducted a one-week experiment, comparing the coding process of qualitative researchers with MAXQDA and Cody when coding a public dataset of interviews. I found that code rules provide both structure and transparency, particularly when coding new data. Explanations for suggestions are commonly desired but rarely used, and perceived quality rather than confidence scores convince users. Finally, working with Cody (for now) benefits coding quality rather than coding speed, increasing the intercoder reliability, calculated with Krippendorff's Alpha, from 0.085 (MAXQDA) to 0.33 (Cody).

4. Part II: Feedback-based Requirements Elicitation [15]

4.1 Study 4: Voice of the Users - Exploring Software Feedback Engagement

4.1.1 Introduction

Software users write online about their applications, often reporting issues they encounter or ways they would like the product to improve. These insights are essential for development teams as they provide requirements to improve their products to satisfy their users better. Organizations want their products to be rated positively since this can help grow their user-base (Pagano & Maalej, 2013). Previous studies have identified requirements information in user feedback on app stores, product forums, and social media (Guzman, Alkadhi, et al., 2016; Guzman, Alkadhi, et al., 2017; Pagano & Maalej, 2013; Tizard, Wang, et al., 2019). This feedback has been called the *voice of the users*, with much recent research studying efficient methods to extract requirement insights (e.g., Guzman, Ibrahim, et al. (2017), Maalej and Nabil (2015), Sorbo et al. (2017), and Tizard, Wang, et al. (2019)).

However, not all software users provide online feedback. If online feedback is being used to drive product development decisions, the concerns and desires of only the vocal users are being considered. If the demographics of the vocal users are not representative of the overall set of users, this introduces the possibility of developing biased software that does not meet the needs of all users. Therefore, it is vital to understand which software users do give online feedback and, in doing so, identify groups whose views may be underrepresented.

However, very little research has investigated who is giving online feedback for software products concerning users' demographics. This may be because demographic information of feedback givers is not readily available. On some feedback channels, even the full name of the person providing the feedback is unavailable. Some preliminary studies have investigated the gender and geographic location of users who provide feedback on app stores (Guzman, Oliveira, et al., 2018; Guzman & Rojas, 2019). These studies found that men were more likely than women to provide feedback on the Apple app store. However, these results are obtained by approximating gender based on usernames since the actual gender identity of the feedback givers is not available on app stores.

In Study IV, I overcome the online data sparsity problem by directly surveying software users about their feedback-giving habits. In an initial survey, I asked 1040 software users about their feedback giving habits on three popular channels: app stores, product forums, and social media[16]. Information on users' demographics and software use was also collected,

[15] This chapter is based on the following studies which are published or in work: Tizard, Rietz, and Blincoe (2020), Tizard, Rietz, Liu, et al. (2021).

[16] *App stores* comprise typical sources of apps, such as the Apple app store or the Google Play Store, where users can provide written feedback and star ratings for apps. *Product forums* are websites separate from store pages and devoted to specific products or companies. *Social media* include outlets such as Facebook, Reddit, Instagram, and allow users to comment and share feedback without special moderation, often on dedicated company pages.

allowing for examining feedback habits across multiple demographics categories (gender, age, education, and ethnicity), finding significant differences in the gender and age of feedback givers. I also investigated what motivates feedback givers and if their software usage habits relate to their feedback giving habits.

In a second survey, I asked 936 software users about why they choose not to give feedback when they face software issues and potential ways they could be encouraged to give feedback. Again, demographic information was collected from respondents, allowing the analysis of differences in feedback behavior between demographic groups.

To get a comprehensive view of online software feedback behavior, I detail RQ4a & RQ4b (as introduced in Section 1.2) into five Sub-RQs that I outline in the following:

Sub-RQ1: *What are the demographics of software users who report giving online written feedback?*

Sub-RQ2: *What motivates software users to give online feedback, and are there differences across demographics?*

Sub-RQ3: *When software does not meet expectations, what are the reasons users decide not to give online feedback?*

Sub-RQ4: *What new methods are perceived to increase the likelihood of software users giving online written feedback?*

Sub-RQ5: *Does the likelihood of giving online written feedback vary based on the type of software used and the duration of software usage?*

The contributions of Study IV are insights about which software users give online feedback, what motivates users when they give feedback, and what discourages them when they do not. Specifically: (1) I show that there are differences in the feedback habits of software users based on traditional demographics. For gender, men reported giving more written feedback than women. With age, distinct patterns emerged, with respondents between 35 and 45 reporting to give the most written feedback on all channels.

(2) I show that user groups have different motivations to give feedback, and these motivations vary across each of the three feedback channels. Respondents also reported differences in the success of in-app prompts between eliciting app ratings and written feedback and differences in the frequency individual feedback givers write on app stores, product forums, and social media.

(3) I present a detailed list of the top reasons for users refraining from giving online feedback. I found the top three reasons to be the same across all three study channels, namely: Looking for an existing answer instead, finding an alternative app instead, and feeling a resolution would take too long. However, there are significant differences in the reasons not to give feedback between channels, between men and women, and between age groups.

(4) I examined user perceptions on new methods to encourage online feedback. I found that users are more encouraged by potential incentive-based elicitation methods such as

in-app rewards compared to possible alternative feedback options like a smart assistant or audio recording. However, many respondents still agreed that alternative options could encourage their feedback.

(5) I present evidence that software users' feedback habits also vary concerning how they use the software. Respondents who spend more hours each day on their phone or computer report giving more written feedback about the software they are using. The software platform being used also relates to written feedback rates, with Linux (computer) and Android (phone) users reporting to give more feedback than those using other platforms.

My findings provide valuable context for requirements elicited from online feedback, identifying underrepresented user demographics. Findings on what motivates and discourages user feedback gives insight into how feedback channels and developers can increase engagement with their user base.

4.1.2 Methodology

I conducted two surveys of software users to answer my research questions, asking about their feedback habits on three channels: app stores, forums, and social media. An initial survey was conducted in December 2019, investigating if reported feedback habits and motivations differed across demographics (*Sub-RQ1, Sub-RQ2, Sub-RQ5*), receiving 1040 complete responses. A second survey was undertaken in November 2020, extending the initial work, investigating the reasons software users do not give online feedback (*Sub-RQ3*), and looking at ways to encourage feedback (*Sub-RQ4*). This second survey received 936 complete responses.

Survey Design

First Survey. The original survey consisted of 24 multiple-choice questions in five main sets, as shown in Table 4.1. The first three sets of questions asked about the feedback the participant provides in the three feedback channels under investigation: app stores (Q1-5), social media (Q6-9), and product forums (Q10-13). The remaining two sets of questions collect software usage information (Q15-18) and demographic information (Q19-24). Descriptions of what was meant by app store and product forum feedback were given within the survey to help participants understand the question context, shown in Figures 4.1 and 4.2. Questions eliciting details on feedback habits were asked before software usage and demographic questions to highlight the study's propose and maintain participant interest.

The sets of questions on the three feedback channels each follow the same general format. First, the participant is asked if they have given feedback on that channel. Next, if applicable, they are asked how frequently they give feedback, the type of feedback given (e.g., reporting a bug), and their motivation for providing feedback on this channel. These questions were all multiple choice. The answer options for the type of feedback provided and the motivation for providing feedback were based on recent research studies on each of these feedback channels. The participants were also asked about their perceptions on the impact of their feedback on influencing changes in the software products (Q14).

Question	Sub-RQ	Topic	Question Text	Answer Source
Q1.	All	App store	What review types have you given to mobile apps in the past? (*choose all that apply*)(*None / Prompted rating / Prompted written review / Direct rating / Direct written review*)	-
Q2.	Sub-RQ2	App store	How many times have you given mobile apps you use a star rating in the last year? (*None / 1-4 times / 5-12 times / 13-26 times / 27-52 times / 53 or more times*)	-
Q3, 7, 11.	All	App store (Q3), Product forum (Q7), Social media (Q11)	How many times have you written (*or given a review*) on ***this channel*** in the last year? (*None / 1-4 times / 5-12 times / 13-26 times / 27-52 times / 53 or more times*)	-
Q4, 8, 12.	Sub-RQ1	App store (Q4), Product forum (Q8), Social media (Q12)	What types of posts (*or reviews*) have you written about software (*or apps*)? (*choose all that apply*) (*Praise (all channels) / Report bug (all channels) / Request feature (all channels)) / Ask a question (all channels) / Recommend to others (app stores, social media) / Dissuade others (app stores, social media) / Discuss shortcoming (app stores, social media) / Dispraise or criticise (app store, product forum) / Discuss a helpful situation (app stores) / Discuss specific feature (app stores) / Assist others (product forums) / Other, please specify (all channels)*)	(Q4) Pagano and Maalej (2013), (Q8) Tizard, Wang, et al. (2019), (Q12) Guzman, Alkadhi, et al. (2017)
Q5, 9, 13.	Sub-RQ2	App store (Q5), Product forum (Q9), Social media (Q13)	What was your motivation(s) to write on ***this channel*** in the past? (*choose all that apply*)(*Show appreciation / Show dissatisfaction / Influence improvement / Recommend / Discourage others / Connect or socialise about software / No specific motivation / Other, please specify)*	(Q5) Pagano and Maalej (2013), (Q9) Tizard, Wang, et al. (2019), (Q13) Guzman, Alkadhi, et al. (2017)
Q6.	All	Product forums	How have you used software product forums in the past? (*choose all that apply*)(*I haven't / Reading and viewing / Written posts*)	-
Q10.	All	Social media	Have you used social media (E.g. Twitter, Facebook) to discuss software products you are using? (*choose all that apply*)(*I haven't / Reading and viewing / Written posts*)	-
Q14.	Sub-RQ2	App store, Product forum, Social media	How likely do you think it is for an app/software product to change based on your online reviews? (*Definitely will / Probably will / Might or might not / Probably won't / Definitely won't*)	Likert (1932)
Q15.	Sub-RQ3	Software usage	What type of mobile phone do you currently use? (*choose all that apply*)(*iPhone / Android (E.g. Samsung, Pixel) / I don't use a mobile phone / Other, please specify*)	-
Q16.	Sub-RQ3	Software usage	What type of computer do you currently use? (*choose all that apply*)(*Windows / Mac (Apple) / Linux / I don't use a computer / Other, please specify*)	-
Q17.	Sub-RQ3	Software usage	How many hours per day do you use your phone? (*Less than 1 hour / 1-4 hours / 4-8 hours / More than 8 hours*)	-
Q18.	Sub-RQ3	Software usage	How many hours per day do you use your computer? (*Less than 1 hour / 1-4 hours / 4-8 hours / More than 8 hours*)	-
Q19.	Sub-RQ1	Demographics	Do you work or have you previously worked in the software industry? (*No / I work or have worked in software / Other, please specify*)	-
Q20.	Sub-RQ1	Demographics	How old are you? (*Under 18 years old / 18-24 years old / 25-34 years old / 35-44 years old / 45-54 years old / Over 55 years old*)	"New Zealand Census" (2018)
Q21.	Sub-RQ1	Demographics	What is your gender? (*Man / Woman / Prefer not to say / Prefer to self-specify (please specify)*)	"New Zealand Census" (2018)
Q22.	Sub-RQ1	Demographics	What is your ethnicity? (White (European) / Asian / Pacific people / African/ Middle Eastern / Latin American / Other, please specify)	"New Zealand Census" (2018)
Q23.	Sub-RQ1	Demographics	What is your highest level of education completed? (*Secondary school / Post secondary, Vocational training / 1-2 year tertiary education / Bachelor degree (3-4 years) / Master degree (postgraduate), Doctoral (postgraduate) / Other, please specify*)	ISCED (2012)
Q24.	Sub-RQ1	Demographics	What is your current employment status? (*Employed full-time (> 40 hours) / Employed part-time (< 40 hours) / Currently unemployed / Student / Retired / Self-employed / Unable to work / Other, please specify*)	"New Zealand Census" (2018)

Table 4.1: First survey questions.

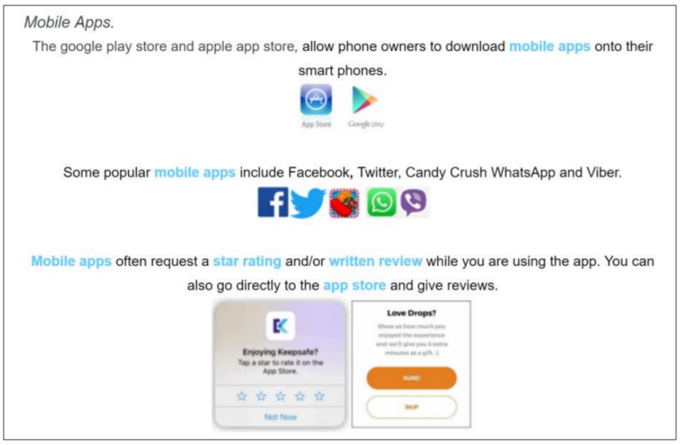


Figure 4.1: Survey mobile app and app store descriptions.

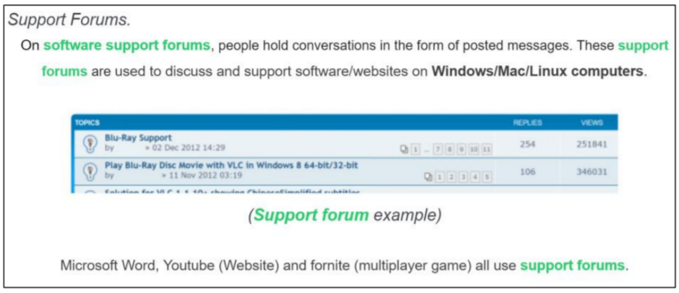


Figure 4.2: Survey support forum description.

For some questions, participants could select more than one answer choice (e.g., motivation for giving feedback). The complete list of questions and answer choices is shown in Table 4.1, with abbreviated answers for each question. An unabbreviated copy of the survey can be found on Zenodo[17].

The software usage questions asked participants how they interact with software products, including the types of devices they use, the types of software they use, and their hours on devices each day. The answer choices for the types of software were obtained from the categories of apps on popular app stores.

The demographic questions collected information on the participants' age, gender, ethnicity, education, and employment. These questions and their associated answer choices were informed by traditional marketing demographic categories (Papadopoulos et al., 2011) as well as the New Zealand census (2018) ("New Zealand Census", 2018).

Second Survey. Analysis of the initial survey showed overall low feedback rates, with underrepresented demographic groups. This prompted a second follow-up survey to understand why users often do not give feedback and how they could be better encouraged to in the future.

For the second survey, four new multi-choice questions (EQ1-4) (see Table 4.2) were added to the first survey's demographic and software usage questions. The new questions were

[17]https://zenodo.org/record/3674076#.XkxNFygzZPY

Question	Sub-RQ	Topic	Question Text	Answer Source
EQ1-3	Sub-RQ3	App store (NQ1), Product forum (NQ2), Social media (NQ3)	*Please rate your* ***agreement level*** *with the following statements:* In the past, when an app/software didn't meet my expectations, I've chosen not to write a review/post because, a) I wasn't aware I could influence app/software improvements by writing a review/post b) I thought it would take too long to get a resolution with a review/post c) I've found ***this channel*** confusing or hard to use d) I didn't think an app/software review/post would be seen by developers or lead to a resolution e) I would look for an existing answer online instead of writing a review/post f) I would look for an alternative app/software instead of writing a review/post g) I didn't think my review would influence other app/software users h) Other reason (please specify)	-
EQ4	Sub-RQ4	App store, Product forums, Social media	*Please rate your* ***agreement level*** *with the following statements:* I would be more likely to post on app stores, forums, or social media about software issues or requests in the future if, a) I would receive a small financial incentive b) I would receive in app rewards. E.g. game currency c) I could give feedback via audio d) I could give feedback via video e) I could give app feedback through a smart assistant (Alexa, Google Assistant) f) Other (please specify)	(Guo & Barnes, 2007), (Turk, 2012), (Stade et al., 2020)

Table 4.2: Second survey questions.

placed before the demographic and usage questions to highlight the survey's focus and encourage engagement. The complete second survey has been made available on Zenodo[18].

The first three questions (EQ1-3) focus on reasons not to give online feedback (Sub-RQ3), asking about each of the three study channels. As this is a new area of software engineering research, there was no existing literature to draw on for answer options. The options for EQ1-3 were primarily sourced from the first survey in-person collection. Participants frequently gave reasons they did not give feedback when asked about their feedback giving habits, including (Table 4.2): *option a*) I wasn't aware I could influence improvements, *b*) it would take too long for a resolution, *d*) it wouldn't lead to a resolution, *e*) I'd find an existing answer instead, *f*) I'd find an alternative app instead. For *option c*, "*The essential guide to user interface design*" Galitz (2007) says confusing interface elements, such as confusing layout or navigation, will quickly lead to user abandonment. Finally, *option g*) was given as an inverted option from the two first survey motivations of recommending or discouraging other users from downloading software, which was cited as motivating by many respondents.

The fourth extension question (EQ4) is focused on new methods to encourage user feedback (Sub-RQ4) across all study channels. This question gives five multi-choice answers, three new methods to give feedback, and two reward types to incentivize feedback. The three new methods to give feedback (audio recording, smart assistant, video recording) were sourced from and inspired by the work of Stade et al. (2020) on smart home feedback.

The reward incentive options are a (small) financial reward and in-app rewards such as

[18]https://zenodo.org/record/4320164#.X9beD9gzZ3g

in-app currency or digital items. Financial incentives have been used effectively in recent years to elicit crowd-sourced data on platforms such as Amazon Mechanical Turk (Turk, 2012). In-app or digital items have shown to have real-world value. Many modern games offer market places where users exchange billions of dollars for digital items (Marder et al., 2019), suggesting digital incentives may also be effective for software feedback.

Recruiting Participants

I used convenience sampling for both surveys to recruit participants (Etikan, 2016). I selected convenience sampling for its usefulness for engaging a high number of participants in a reasonable period. The possible sources of bias from my sampling methodology are discussed in section 5.3. As an incentive for survey participation, I offered each participant a chance to join a raffle to win a \$200/€120 cash prize. The survey was primarily made available online through the Qualtics survey platform ("Qualtrics", 2019).

First Survey. Participants were invited via a link to the Qualtics survey distributed on Facebook and Twitter. In addition, I recruited from a pool of university participants using the hroot software (Bock et al., 2012). The pool includes nearly 3500 participants who registered online to be invited to and participate in scientific studies, either on-site or online. This pool was mainly advertised at the Karlsruhe Institute of Technology, so the pool primarily contains students between 18 and 30. Through hroot, 2570 participants were invited. Hardcopies of the survey were also distributed in public areas of Auckland city during December 2019. The completed hardcopy survey responses were manually consolidated with the online survey responses. The survey was open to anyone 16 years or older.

Second Survey. The second survey was also hosted on Qualtrics. Once again, participants were contacted through the hroot software pool, recruited from the Karlsruhe Institute of Technology. Additionally, participants from the first survey, who indicated they would like to receive the study results, were invited to participate in the second survey when the results were sent. About 1300 participants were invited through hroot for the second survey.

Furthermore, I recruited new participants for the second survey through Zhejiang University, China. The survey was advertised in Zhejiang University's online student forums (CC98 and Duoduo Xiaoyou), with respondents being given a chance to win one of several ¥200 prizes as a substitute to the \$200 prize offered in New Zealand and Germany. For the Zhejiang University distribution, the second survey was translated from English to Mandarin by a paid contractor and was then reviewed by a native Mandarin speaker before distribution. The translated survey has been made available on Zenodo[19]. Open-ended responses were translated back to English for analysis.

Survey Participants

First Survey. Across all collection channels, 1040 participants fully completed the survey. All respondents reported having used software on a computer or mobile. Therefore all

[19]https://zenodo.org/record/4320182#.X9bmt9gzZ3g

Demographic Type	Group	First Survey Respondents	Second Survey Respondents
Gender	Men	571 (*54.9%*)	500 (*53.5%*)
"	Women	454 (*43.7%*)	418 (*44.7%*)
"	Gender diverse	16 (*1.5%*)	18 (*1.9%*)
Age	Under 18 years	61 (*5.9%*)	7 (*0.8%*)
"	18 - 24 years	571 (*54.9%*)	629 (*67.2%*)
"	25 - 34 years	285 (*27.4%*)	270 (*28.9%*)
"	35 - 44 years	50 (*4.8%*)	24 (*2.6%*)
"	45 - 54 years	29 (*2.8%*)	5 (*0.5%*)
"	Over 55	44 (*4.23%*)	1 (*0.11%*)
Ethnicity	White/European	790 (*76.0%*)	415 (*44.3%*)
"	Asian	149 (*14.3%*)	463 (*49.5%*)
"	Middle Eastern	26 (*2.5%*)	14 (*1.5%*)
"	Latin American	24 (*2.3%*)	13 (*1.4%*)
"	Pacific and Maori	18 (*1.7%*)	3 (*0.3%*)
"	African	7 (*0.7%*)	7 (*0.7%*)
"	Other	27 (*2.6%*)	21 (*2.2%*)
Education	Secondary school	411 (*39.5%*)	184 (*19.7%*)
"	Vocational Training	14 (*1.4%*)	6 (*0.6%*)
"	1-2 year Tertiary	62 (*5.9%*)	18 (*1.9%*)
"	Bachelor degree	390 (*37.5%*)	515 (*55.0%*)
"	Master degree	129 (*12.4%*)	183 (*19.6%*)
"	Doctoral degree	25 (*2.4%*)	25 (*2.7%*)
"	Other	9 (*0.9%*)	5 (*0.5%*)
Employment	Full time (> 40 hours)	215 (*20.7%*)	119 (*12.7%*)
"	Part time (< 40 hours)	78 (*7.5%*)	35 (*3.7%*)
"	Student	644 (*61.9%*)	750 (*80.1%*)
"	Self-employed	28 (*2.7%*)	5 (*0.5%*)
"	Currently unemployed	39 (*3.8%*)	10 (*1.1%*)
"	Retired	15 (*1.4%*)	2 (*0.2%*)
"	Unable to work	4 (*0.4%*)	0 (*0.0%*)
"	Other	18 (*1.7%*)	15 (*1.6%*)

Table 4.3: Respondent demographics.

respondents are software users. The make up of the survey respondents regarding gender, age, ethnicity, education, and employment is shown in Table 4.3.

Regarding the highest level of education obtained, I noticed that many respondents reported secondary school (411) and bachelor's degree (390). Given the hroot software recruited from a pool of university participants, I suspected education level could be associated with the age of the participants. I saw that 90.02% of secondary school educated reported to be under 25, compared to only 41.61% of those who have higher education. After controlling for age, I did not see any significant differences in feedback habits regarding education level. Thus, I do not report results considering education level.

Second Survey. Across all the collection channels, 936 participants fully completed the extension survey. The sample comprises 423 respondents recruited through Zhejiang University, 420 respondents through the Karlsruhe Institute of Technology pool, and 93 respondents invited through the first survey follow-up. A demographic breakdown of the extension survey respondents is shown in Table 4.3.

Survey Analysis

To answer my research questions, I analyzed the ratio of respondents in each user group (based on demographics or software usage) that reported a particular behavior, e.g., giving feedback on a particular feedback channel or having a specific motivation. Chi-squared tests, which tests for differences in proportion between two groups (McHugh, 2013), were used to find if differences in reported behaviors between user groups are statistically significant.

Statistical significance (chi-squared) was calculated for Likert scales answers by considering *strongly agree* and *agree* as a single agreement value. Likewise, *strongly disagree* and *disagree* were combined as a single disagreement value, with neutral values not used in the calculation.

Optional open-ended answers, in addition to the primary closed-ended options, were given for motivation to give feedback (Sub-RQ2), reasons not to give feedback (Sub-RQ3), and methods to encourage feedback (Sub-RQ4). These open ended-responses have been categorized into common themes using Thematic Content Analysis (Braun & Clarke, 2006). Themes are presented with a typical example and the number of responses in the theme. Findings and codes of the thematic content analysis were discussed and iterated with other contributors to this research study.

Concerning gender, the majority of participants identified as men or women. I did give participants the option to self-specify gender. However, too few participants chose this option in order to find statistically significant results. Thus, my analysis was limited to only the differences between participants who identified as men and women.

4.1.3 Results

4.1.3.1 Demographics

Sub-RQ1: *What are the demographics of software users who report giving online written feedback?*

In this section, I present the percentage of *written* feedback givers in each demographic group.

Feedback across online channels. Overall, 30.96% of survey respondents reported having given written feedback on any of the three online channels. Most survey respondents reported having written feedback on app stores (18.16%), then on product forums (13.45%),

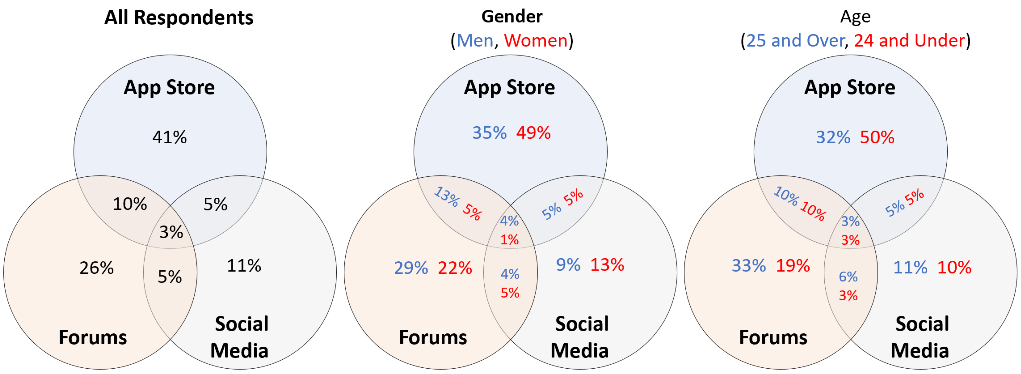


Figure 4.3: Overview - The feedback given to each online channel, as a proportion of respondents who had given online written feedback.

and least on social media (7.11%). The majority of feedback giving respondents gave feedback to only one channel (77.64%), 19.57% had written on two channels, with 2.80% writing on all three (Figure 4.3). A Chi-squared test showed the higher rate of respondents using only one feedback channel over multiple channels is statistically significant ($p<0.001$).

Age. Under 18's, reported to have given the least feedback of all ages, across all channels (*app store 6.6%, forums 0.0%, social 4.9%*) (Figure 4.4).

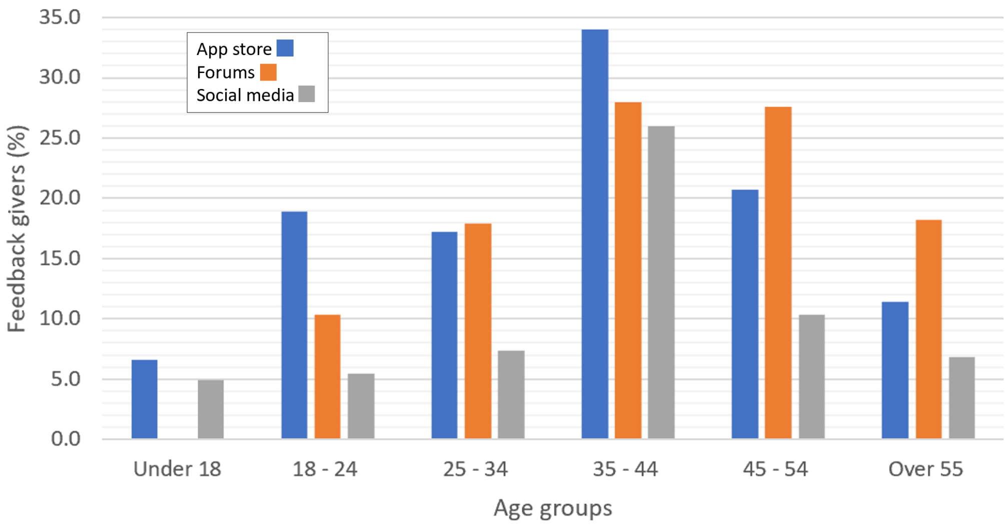


Figure 4.4: User feedback with age.

Respondents under 25 preferred to give feedback to the app store over product forums, shown in bold in Table 4.4. Under 25's preference for app stores was shown to be statistically significant using a chi-squared test ($p < .001$). Respondents 25 and over used app stores and forums more equally, with those over 44 reporting more forum use. However, the differences in channel use for those 25 and above were not found to be significant.

Conversely, 35-45 year old's (50 respondents), reported to give the most feedback across all channels (*app store 34.0%, forums 28.0%, social 26.0%*). Chi-squared tests show there are statistically significant differences between ages (shown in Table 4.5).

	App Store (%)	**Product Forums (%)**
Under 25 years old	17.72	9.34
25 years old and over	18.87	19.85

Table 4.4: Comparing app store and forum feedback with age.

	App Store		Product Forums		Social Media	
Compared Age Groups	*Chi2*	*p*	*Chi2*	*p*	*Chi2*	*p*
Under 18 < 18 - 24	**4.96**	**0.026**	**5.79**	**0.016**	0.017	0.896
Under 18 < 25 - 34	3.600	0.058	**11.419**	**0.001**	0.165	0.685
Under 18 < 35 - 44	**11.760**	**0.001**	**17.087**	**< 0.001**	**8.264**	**0.004**
18 - 24 < 25 - 34	0.27	0.603	**9.04**	**0.003**	0.936	0.333
18 - 24 < 35 - 44	**5.603**	**0.018**	**12.183**	**< 0.001**	**26.509**	**< 0.001**
25 - 34 < 35 - 44	**6.570**	**0.01**	2.169	0.141	**14.214**	**< 0.001**
35 - 44 > 45 - 54	0.997	0.318	0.049	0.825	1.90	0.168
45 - 54 > Over 55	0.571	0.450	0.437	0.508	0.010	0.919
35 - 44 > Over 55	**5.487**	**0.019**	0.770	0.380	**4.815**	**0.028**

Note: statistically significant results are bolded

Table 4.5: User feedback with age.

Gender. Men reported to give more feedback than women across all channels, shown in Table 4.6. On apps stores, 20.3% of men and 14.5% of women reported giving feedback. On product forums, 18.0% of men and 8.1% of women reported giving feedback. On social media, the difference was the smallest, with 8.2% and 5.7% respectively reporting to give feedback. Chi-squared tests showed that the difference between men and women respondents was statistically significant for app stores (p=*0.02*) and product forums (p<*0.001*).

Men and women respondents reported some differences in the types of feedback they give on all three feedback channels, shown in Table 4.7. More women feedback givers reported praising apps on app stores than feedback giving men (*w: 50%, 41.38%*) and also reported giving bug reports. More men reported describing a situation an app was helpful, reported a shortcoming of an app, and requested new features.

On product forums, both men and women were very likely to ask a question about the

	Number of Respondents	**App Store (%)**	**Product Forums (%)**	**Social Media (%)**
Men	571	20.32	18.04	8.23
Women	454	14.54	8.15	5.73

Table 4.6: User feedback with gender.

	App Store (%)		Forums (%)		Social Media (%)	
Feedback Type	Men	Women	Men	Women	Men	Women
Praise	41.38	50.00	20.39	10.81	38.30	30.77
Report bug	40.52	48.48	73.79	56.76	46.81	42.31
Request feature	26.72	18.18	32.04	21.62	27.66	38.46
Ask question	2.59	6.06	88.35	94.59	68.09	65.38
Recommend to others	12.96	16.67	*NA*		36.17	19.23
Dissuade others	10.34	6.06	*NA*		8.51	11.54
Discuss shortcomings	47.41	36.36	*NA*		46.81	34.62
Dispraise or criticise	18.10	15.15	16.50	8.11	*NA*	
Helpful situation	36.21	27.27	*NA*		*NA*	
Discuss feature	21.55	22.73	*NA*		*NA*	
Assist others	*NA*		55.34	21.62	*NA*	

Table 4.7: User feedback type with gender.

software, with 88.35% of men feedback givers and 94.59% of women. Men feedback givers were more likely to give other types of feedback, including: report a problem, request a feature, give praise, give criticism and assist others. On social media, more men reported recommending software to others and discussing shortcomings. More women reported requesting new features.

Employment. Respondents working full time reported using product forums at a higher rate than those working part time and students (Table 4.8). However, there is a strong association between employment level and age as 78.57% of students are also under 25. In the bottom half of Table 4.8, all under 25-year-old respondents were removed from the analysis, showing the difference between employment levels is not as prominent when considering only older respondents. The feedback differences between employment groups were not statistically significant, using chi-squared tests, after excluding the under 25-year-old respondents.

Software professionals. Respondents who work, or have worked in software (software professionals), reported to have given feedback at a higher rate than those who have not worked in software on all channels (Table 4.9). Chi-squared tests showed that the feedback

	Number of Respondents	Forums (%)	Under 25 years (%)
Full-time	215	21.40	20.93
Part-time	78	12.82	66.67
Student	644	10.87	78.57
Full-time (*no under 25's*)	170	22.94	0.00
Part-time (*no under 25's*)	26	11.54	0.00
Students (*no under 25's*)	138	18.84	0.00

Table 4.8: User Feedback with employment type.

	Number of Respondents	App Store (%)	Forums (%)	Social Media (%)
Software Professionals	171	27.49	19.88	12.87
Other Respondents	869	16.32	12.18	5.98

Table 4.9: Feedback of software professionals.

rate difference between software professionals and other respondents was significant on all channels (app stores: $p=0.001$, product forums: $p=0.01$, social media: $p=0.002$).

Ethnicity. The majority of survey respondents were either Caucasian (790) or Asian (149), which limited my findings with respect to ethnicity. However, the ethnic demographics of the respondents are representative of a study based in New Zealand and Germany. Only the difference between Caucasian and Asian feedback rates could be investigated, and this difference was not statistically significant on any channel.

Answer to Sub-RQ1. There are statistically significant differences in the amount of written feedback given by software users concerning traditional demographics. For gender, men reported giving more feedback than women on all three feedback channels. The types of feedback men and women reported giving also varied in unique ways. With age, distinct patterns emerged, with respondents between 35 and 45 reporting to give the most feedback and under 18's reporting to give the least, on all channels. Additionally, software professionals reported giving significantly more feedback than other respondents.

4.1.3.2 Motivations

Sub-RQ2: *What motivates software users to give online feedback, and are there differences across demographics?*

The section presents my findings concerning what motivates users to give online feedback. Therein, I outline the difference in motivations across the three channels and between groups.

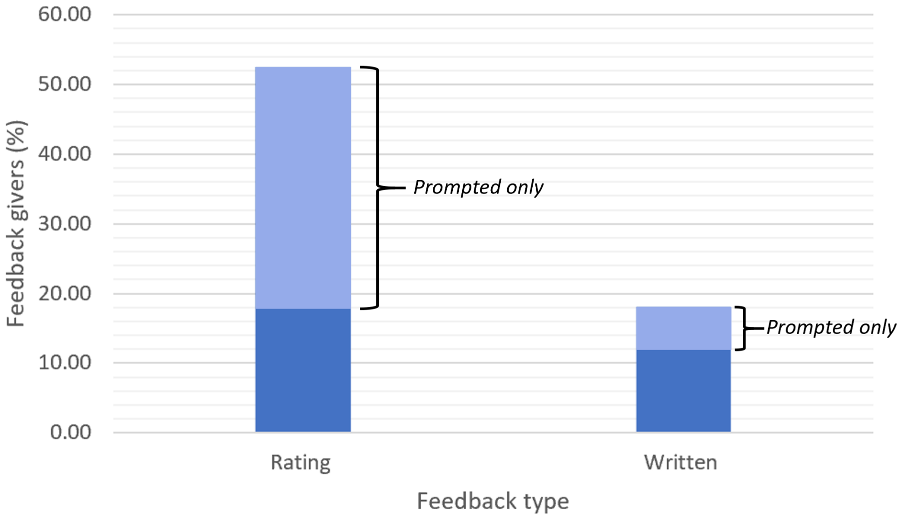


Figure 4.5: Impact of in app prompts.

Overall. The reported motivations to give feedback on app stores, product forums and social media are given in Table 4.10. I report the motivations as a percentage of all users who give written feedback on each channel. Each respondent could state multiple motivations. As can be seen, the motivations vary across feedback channels. Show appreciation for software was the most commonly cited motivation on app stores (65.15%) and Social media (56.76%). Get help with software was the top motivation to post on product forums (70.37%). Influencing improvement was also a major motivation, being the third most cited on all channels.

App Store	(%)	Product Forum	(%)	Social Media	(%)
1. Show appreciation	65.15	1. Get help	70.37	1. Show appreciation	56.76
2. Influence improvement	52.02	2. Influence improvement	44.29	2. Influence improvement	51.35
3. Show dissatisfaction	34.85	3. Show appreciation	26.43	3. Show dissatisfaction	37.84
4. Recommend to others	29.80	4. Recommend to others	17.86	4. Connect or socialise	35.14
5. Discourage others	12.63	5. Show dissatisfaction	16.43	5. Recommend to others	32.43
6. Get help	9.20	6. Connect or socialise	15.72	6. Get help	22.73
7. No specific motivation	5.05	7. No specific motivation	7.86	7. Discourage others	14.86
8. Connect or socialise	1.52	8. Discourage others	3.57	8. No specific motivation	8.11

Table 4.10: Motivations to give feedback.

Mobile app prompts. 52.45% of all survey respondents reported having previously given a star rating to an app (Figure 4.5). Of those who have given a star rating, 65.75% only gave the rating when prompted within the app, never directly on the app store. 18.16% of respondents reported having given a written review to an app. Of those who have given a written review, 31.75% only gave a written review when prompted to within the app.

Gender. Some differences in motivations to give feedback were reported between men and women. The percentage of men and women feedback givers who cited each motivation are

	App Store (%)		Product Forums (%)		Social Media (%)	
Motivation	Men	Women	Men	Women	Men	Women
Show appreciation	67.24	72.73	28.16	18.92	57.45	53.85
Show dissatisfaction	36.21	36.36	13.59	21.62	31.91	46.15
Influence improvement	57.76	50.00	49.51	27.03	55.32	42.31
Recommend	29.79	34.62	32.76	30.30	18.45	13.51
Discourage	16.38	6.06	1.94	5.41	10.64	19.23
Connect/ socialise	4.31	7.58	13.59	18.92	27.66	42.31
Get help	10.14	5.88	71.11	77.78	41.18	0.00
No specific motivation	0.86	3.03	9.71	0.00	4.26	11.54

Table 4.11: Motivations to give feedback with gender.

shown in Table 4.11. On app stores, men were more motivated to discourage others from using a disliked app. On product forums, more men cited influencing an improvement in the software as a motivation. More women were motivated to show dissatisfaction and connect or socialize about a software product on social media. Also, on social media, more men cited influence improvement and get help. These results are bolded in Table 4.11.

Feedback frequency. The majority of feedback givers reported having given feedback between zero and four times in the last year, across all channels (Table 4.12. App stores had the least respondents reporting to give more than four pieces of feedback. Product forums had the most respondents giving feedback more than four times.

Perception of influencing developers. Survey respondents who believed that software developers would *definitely not* be influenced by online feedback were less likely to give feedback than those who believed influence was more likely, on all channels. However, chi-squared tests showed that these differences were not statistically significant. Feedback rates with the perception of influencing developers are shown in Table 4.13.

	App Store (%)	Forums (%)	Social media (%)
0 to 4	87	12	1
5 to 12	62	30	8
13 or more	71	17	12

Table 4.12: Feedback given by individual users each year, on each channel.

	Number of Respondents	App Store (%)	Forums (%)	Social media (%)
Definitely will	83	14.46	18.07	7.23
Probably will	265	19.25	13.96	7.92
Might or might not	416	18.75	14.90	6.49
Probably will not	248	18.95	9.68	8.06
Definitely will not	27	3.70	7.41	0.00

Table 4.13: User feedback with perception of influencing developers.

Other motivations. Some survey respondents offered additional motivations when they were asked what motivates them to give app store feedback, in an optional open-ended response field. I categorized the open responses into two themes. The most commonly reported other motivation to give app feedback was to ***Get rid of the feedback prompt***, with 20 related responses. One respondent said *"The number of times they asked me to rate it was getting annoyingly high so I just did it so they would stop prompting me"*, and another said that they were *"annoyed by the disturbance: hope that no more ratings will be asked after one rating was given"*.

The other theme identified was to ***Receive in-app rewards***, with seven related responses. For example, one respondent said *"you get coins/free stuff if you rate the app sometimes"* and another said they were motivated by *"In-app benefits from Rating the app."*

Answer to Sub-RQ2. Showing appreciation was the top motivation given to write feedback on app stores and social media. On product forums, getting help was the most commonly cited motivation (Table 4.10). Differences in the motivations of men and women to give written feedback on each channel were also reported.

In-app prompts were reported to be very effective at motivating app users to give star ratings but less effective at eliciting written feedback. Individual survey respondents reported engaging with each feedback channel at different frequencies, writing on product forums the most times a year and least on app stores.

4.1.3.3 Reasons Users do not Give Online Feedback

Sub-RQ3: *When software does not meet expectations, what are the reasons users decide not to give online feedback?*

Respondents were asked to rate their agreement on a five-point Likert Scale, with seven predefined reasons that they did not give feedback in the past (Table 4.2), when faced with software issues.

Overall. Respondents reported the same top three reasons not to give online feedback for all three study channels, though the order varied across the channels (see Figure 4.6). The top three reasons were: (1) Users would look for an existing answer online instead of

	App Store			Product Forums			Social Media		
	Men (%)	Women (%)	*Chi2* (*p*)	Men (%)	Women (%)	*Chi2* (*p*)	Men (%)	Women (%)	*Chi2* (*p*)
Alternative app	76	80	1.65	46	43	0.71	67	73	1.20
Existing answer	**74**	**82**	**8.49** (**)	67	57	0.07	79	80	0.82
To long	**75**	**81**	**5.45** (*)	52	45	0.10	55	62	3.17
No resolution	48	53	2.66	30	32	3.64	55	52	0.23
Not aware	**41**	**55**	**21.32** (***)	28	30	3.56	**51**	**56**	**4.47** (*)
Won't influence	28	32	0.01	20	19	0.15	**43**	**34**	**5.44** (*)
Confusing	**17**	**22**	**7.65** (**)	21	20	0.35	14	12	0.58

*Note: statistically significant results are bolded *** $p<0.001$, ** $p<0.01$, * $p\leq 0.05$*

Table 4.14: Reasons not to give feedback, agreement level by gender.

giving feedback. (2) Users would try to find an alternative app instead of giving feedback. (3) Users felt a resolution to their problem would take too long, and therefore would not give feedback. On forums and social media, finding an existing answer had the highest agreement from respondents (83%, 79%). On app stores, all three top reasons had an agreement level of 78%.

Respondents most commonly reported not being aware their feedback could influence software improvements on social media (54%), then on app stores (48%), and least commonly on forums (38%). Forums were most commonly agreed to be confusing or hard to use (28%), then app stores (19%), with social media the least reported to be confusing (13%).

Gender. Differences in the reasons not to give online feedback were reported between men and women on all three channels. More women reported not being aware they could influence software improvements with feedback and found app stores confusing or complicated to use on app stores. Women also reported being more likely to look for existing answers and believe a resolution to their issue would take too long. All these results were statistically significant and have been bolded in Table 4.14.

On forums, men reported more often that they would look for an existing answer instead of giving feedback. However, this was not statistically significant. On social media, women more often reported not being aware they could influence software improvements. Men more often reported not to give social media feedback because they felt it would not influence other users. Both these social media results were found to be statistically significant and have been bolded in Table 4.14.

Age. Differences in the reasons not to give online feedback were reported between those under 25 and those 25 and over. More under 25's agreed that app stores are confusing or hard to use. Under 25's also more commonly reported to feel their app store feedback would not be seen or lead to a resolution. Those 25 and over more often agreed that they would not give feedback because they could not influence other users. These results were found to be statistically significant, as shown in Table 4.15.

On forums, significantly more under 25's agreed they were not aware their feedback could

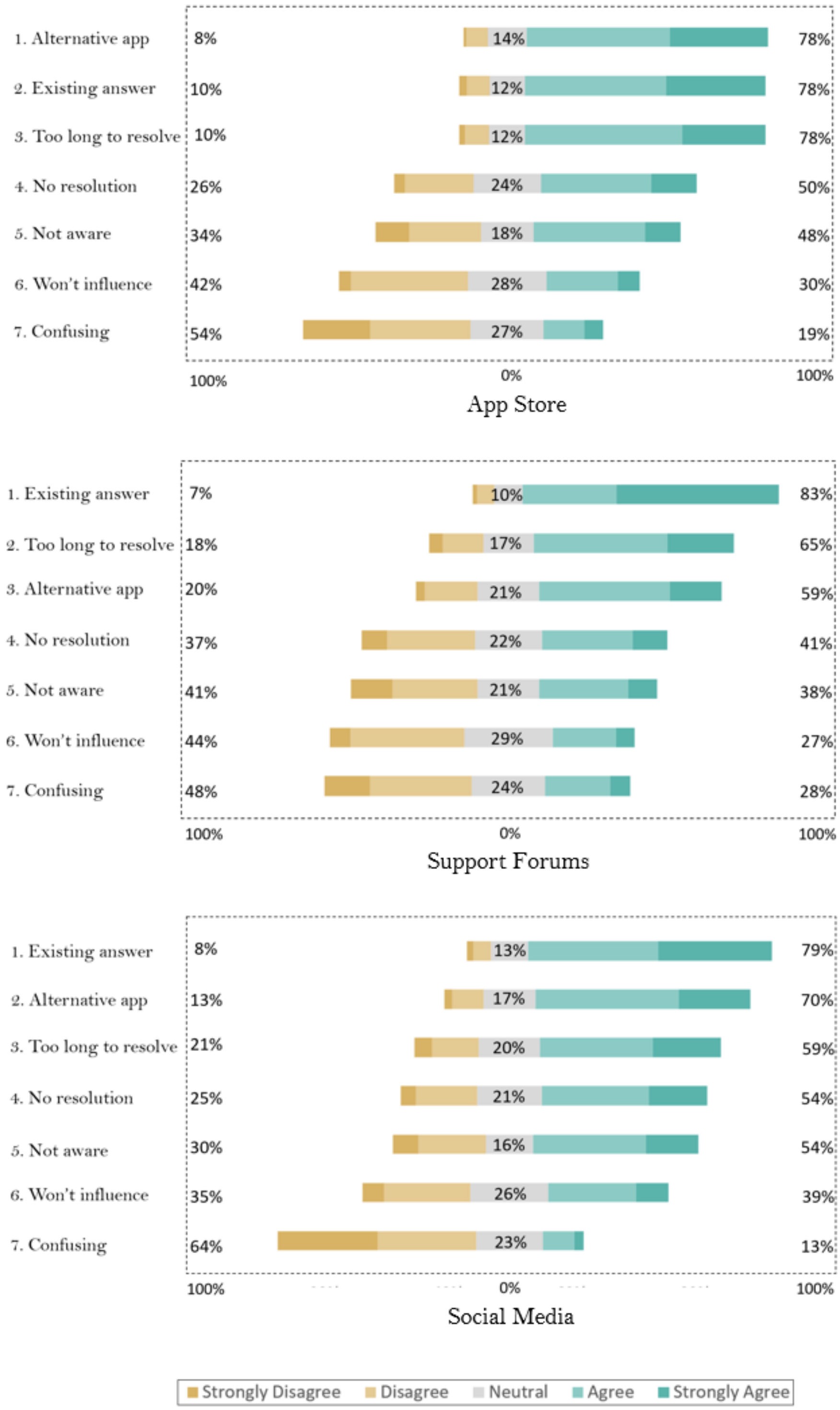


Figure 4.6: Reasons not to give online feedback. Likert scales.

influence software improvements. Under 25's also more commonly reported to feel their forum feedback would not be seen or lead to a resolution. On social media, under 25's more commonly agreed that a resolution to their software issues would take too long and therefore would not give feedback (62%, 52%). These results were all statistically significant (Table 4.15).

	App Store			Product Forums			Social Media		
	25 and over (%)	Under 25 (%)	*Chi2 (p)*	25 and over (%)	Under 25 (%)	*Chi2 (p)*	25 and over (%)	Under 25 (%)	*Chi2 (p)*
Alternative app	79.0	77.7	0.006	60.4	58.6	0.071	70.7	69.2	<0.001
Existing answer	78.0	77.8	0.016	85.5	82.0	<0.001	80.7	78.3	2.096
To long	76.7	78.3	0.002	67.1	63.9	0.115	**51.7**	**61.8**	**8.634** (**)
No resolution	**46.3**	**52.4**	**7.071** (**)	**36.9**	**43.0**	**5.335** (*)	49.3	55.5	1.888
Not aware	41.3	50.5	2.615	**28.2**	**43.9**	**19.656** (***)	50.7	55.0	0.949
Won't influence	**34.7**	**28.0**	**5.489** (*)	26.7	26.5	0.03	33.0	41.5	0.923
Confusing	**13.7**	**21.9**	**8.728** (**)	28.6	26.9	0.001	15.3	12.3	2.511

*Note: statistically significant results are bolded *** p<0.001, ** p<0.01, * p≤0.05*

Table 4.15: Reasons not to give feedback, agreement level by age.

Other reasons. Some survey respondents offered other reasons they do not give feedback, in an optional open response field. These other reasons have been categorised into themes for each feedback channel, with themes cited by at least two respondents shown in Table 4.16.

On app stores, the most common other reason given not to provide feedback was, ***Too much effort required***, with 51 related responses. Two typical responses were, *"It would take too long to write a review"* and *"It seems like to much of a hassle"*. ***Wanting to stay anonymous***, was the second most cited other reason on the app store, with seven related responses.

On forums, the most common other reason not to give feedback (ten responses) was that the respondents ***Don't want to create an account***. One respondent said *"In most cases, you have to create an account for the forum, which makes it more difficult and time-consuming to generate a post"*. ***Wanting to stay anonymous*** was also a barrier to feedback on forums, with five related responses.

On social media, the most common other reason not to give feedback was that respondents ***Don't use social media*** (25 responses). Twelve respondents said they ***Won't post software issues*** on social media, one saying *"I don't want that my close friends and colleagues see such posts of mine."*. ***Wanting to stay anonymous*** when reporting software issues on social media was given by nine respondents.

Answer to Sub-RQ3. Looking for an existing answer, finding an alternative app, and feeling a resolution would take too long were the top three reasons not to give feedback across all three study channels (see Figure 4.6). Between channels, most respondents reported not being aware they could influence software improvements on social media, and Forums were most commonly reported to be complicated or confusing to use. Significant differences in the reasons not to give feedback were also reported between men and women (Table 4.14), and between age groups (Table 4.15). Common other reasons not to give feedback include: Wanting to stay anonymous, not wanting to create an account (on forums), and not wanting to post software issues on social media.

Channel	Summary	Example	Number of Respondents
App store	Too much effort required	*"It would take too long to write a review!"*	51
"	Want to stay anonymous	*"I try to stay anonymous."*	7
"	Too many reviews already	*"The fact that there are so many reviews online on the app store was a reason that I thought it would change nothing to write another review"*	3
"	Avoid having bad influence	*"Don't want to have a bad influence due to my bad review"*	2
"	In-app feedback	*"When the review request showed up, I thought I could direct write a review. But when I clicked the button, I had to use the App store to write the review".*	2
Forums	Don't want to create account	*"I don't want to make an account"*	10
"	Too much effort	*"Can't be bothered"*	5
"	Want to stay anonymous	*"I'm aware that anything I post online could be used against me, even in the distant future."*	4
"	May look bad	"Look bad if already asked"	2
"	Faster channel instead	*"Id rather use any other support method such as email or chat or phone because I feel they respond faster to that"*	2
Social media	Don't use social media	*"I dont use social media"*	25
"	Wont post software issues	*"I don't want to share my support request in social media"*	13
"	Want to stay anonymous	*"I want to stay anonymous"*	9
"	Don't post on social media	*"I generally don't post on social media."*	6
"	Too much effort	*"I was too lazy"*	3

Table 4.16: Other reasons to not give online feedback.

4.1.3.4 Methods to Encourage Online Feedback

Sub-RQ4: *What new methods are perceived to increase the likelihood of software users giving online written feedback?*

Respondents were asked to rate their agreement, on a five-point Likert Scale, with five predefined potential new methods to encourage their feedback (Table 4.2).

Overall. A small financial incentive was the most agreed method to increase respondents probability of giving online software feedback (82%) (Figure 4.7). Next, in-app rewards were thought to be potentially effective, with 65% agreement.

Three alternative methods for giving online feedback were not seen as being as effective at encouraging feedback. Feedback through a smart assistant was most favoured of these (25%), then the option to give feedback via audio recording (17%) and least agreement was given to feedback via video recording (11%).

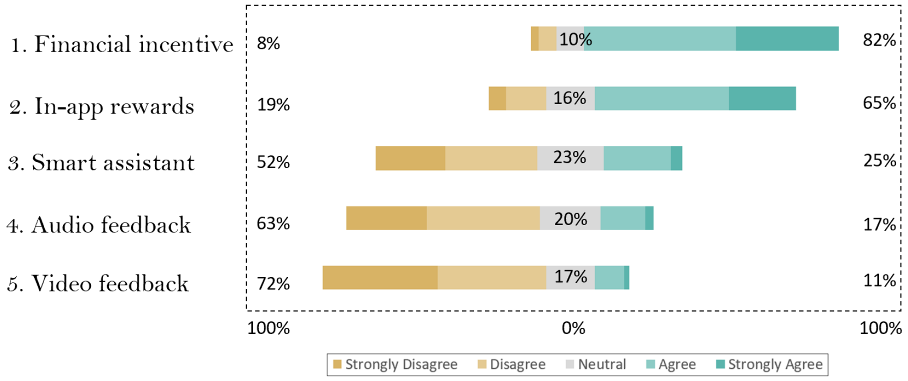


Figure 4.7: Methods to encourage online feedback. Likert scale.

Gender, Age and Employment. More women than men agreed that the ability to give feedback through a smart assistant would increase their probably of giving feedback, which was found to be statistically significant (Table 4.17). The four other proposed methods' agreement level was very similar between men and women and was not statistically significant. With age and employment, the perceived effectiveness of the potential incentives was also similar between groups, with the differences not found to be statistically significant.

	Men	Women	*Chi2*	*p*
Financial incentive	82.6	82.8	0.001	0.981
In-app rewards	65.6	65.6	0.468	0.494
Smart assistant	**22.8**	**28.7**	**6.01**	**0.014**
Audio feedback	18.4	15.6	0.840	0.359
Video feedback	11.4	11.0	0.018	0.893

Note: statistically significant results are bolded

Table 4.17: Methods to encourage feedback, agreement level by gender.

Other methods. Some survey respondents offered other methods to encourage their feedback, in an optional open response field. I categorized these *other* encouragement methods into themes, and those with at least two responses are shown in Table 4.18. The most commonly suggested encouragement method (18 responses) was that users ***Want a quick response*** to show that developers had seen the feedback. One respondent said they would be encouraged if "*I get better feedback like they saw my feedback and are trying to solve my problem*", another respondent said *"I would do it if I know for sure that I will get an answer"*.

Being able to ***Give anonymous feedback*** was said to be encouraging by seven respondents. Moreover, five respondents said that seeing a ***Clear track record*** of developers addressing feedback would increase their likelihood of giving feedback. One respondent said they would be encouraged if *"I saw others making a difference with their suggestions"*, another respondent said *"If there were lists linked on the store/forum/social media which showed on which improvements the developers are working"*.

Summary	Example	Number of Respondents
Want a quick response	*"I could get a response to my review immediately"*	18
Give anonymous feedback	*"The feedback would be completely anonymous."*	7
Clear track record	*"Show a track-record of implemented stuff from reviews"*	5
Feedback to a human	*"I could give feedback in a conversation with a human"*	4
In-app feedback	*"I could give feedback within the app"*	3
Easier to post	*"If posting was easier"*	3

Table 4.18: Other new methods to encourage online feedback, all channels.

Answer to Sub-RQ4. Respondents saw financial and in-app rewards as better methods to encourage feedback than new options such as giving feedback through a smart assistant (Figure 4.7). Women, compared to men, more commonly felt an option to give feedback through a smart assistant would be encouraging (Table 4.17). Respondents suggested additional methods to encourage feedback in open-ended responses. The most common suggestions were: Wanting a quick response to show that feedback had been seen; The ability to give anonymous feedback; And, showing a clear track record of user feedback being addressed.

4.1.3.5 Type of Software and Duration of Use

Sub-RQ5: *Does the likelihood of giving online written feedback vary based on the type of software used and the duration of software usage?*

Concerning computer and phone type, survey respondents were asked to select all device types they use. Therefore, respondents could be counted in multiple categories (e.g., Android and iPhone). For phones, 1.44% (15) of respondents reported using both Android and iPhone. For computers, 10.38% (108) of respondents reported using more than one computer type, with dual use of Windows and Linux being the most common combination (4.90%).

iPhone/Android. Android users reported giving feedback to the app store at a higher rate than iPhone users (Table 4.19). 13.48% of iPhone users reported having given written feedback on app stores compared to 21.84% of Android users. A chi-squared test showed this difference to be statistically significant, given in Table 4.20.

Windows/Mac/Linux. Linux users reported giving written feedback on app stores and product forums at a higher rate than Windows and Mac users (Table 4.19). Chi-squared tests showed these differences to both be statistically significant (Table 4.20). The difference between Windows and Mac users' feedback was not statistically significant.

Device	Number of Respondents	App Store (%)	Product Forum (%)	Social Media (%)
Android	618	21.84	13.75	6.80
iPhone	423	13.48	12.77	7.33
Linux	94	31.91	26.60	10.64
Windows	759	19.10	14.6	6.46
Mac	275	16.73	12.36	10.18

Table 4.19: User feedback with device type.

	App Store		Product Forums	
	Chi2	*p*	*Chi2*	*p*
Android > iPhone	**11.144**	**0.001**	0.087	0.769
Linux > Windows	**7.651**	**0.006**	**8.073**	**0.004**
Linux > Mac	**8.974**	**0.003**	**9.531**	**0.002**

Statistically significant results are bolded

Table 4.20: User feedback with device type. Significance tests.

Hours of computer use. Respondents who reported a higher daily computer use (hours) were more likely to give feedback to product forums. The least forum feedback was given by respondents using their computer less than 1 hour or between 1 and 4 hours a day. Those using their computer between 4 and 8 hours gave more feedback, and those using their computer more than 8 hours a day gave at the highest rate. Chi-squared tests showed that the feedback rate differences between 1 - 4 hours and 4 - 8 hours and between 1- 4 hours and over 8 hours were statistically significant (Table 4.22).

Hours of phone use. Respondents who reported a higher daily phone use (hours) were more likely to give feedback to social media. However, chi-squared tests showed that these differences were not statistically significant.

Daily Computer Use	Number of Respondents	App Store (%)	Forums (%)	Social Media (%)
Less than 1 hour	109	18.35	10.09	6.42
1 - 4 hours	436	15.14	9.40	5.96
4 - 8 hours	363	20.66	17.08	7.99
More than 8 hours	110	21.82	23.64	9.09

Table 4.21: User feedback with daily computer use.

Daily Computer Use	***Chi2***	***p***
Less than 1 hour < 1-4 hours	0.001	0.971
Less than 1 hour < 4-8 hours	2.619	0.106
1 - 4 < Over 8 hours	**15.233**	**< 0.001**
Less than 1 < Over 8 hours	**6.221**	**0.013**
1-4 hours < 4-8 hours	**9.722**	**0.002**
4-8 hours < Over 8 hours	1.983	0.159

Statistically significant results are bolded

Table 4.22: Computer daily use. Significance tests (product forums).

Daily Phone Use	**Number of Respondents**	**App Store** (%)	**Forums** (%)	**Social Media** (%)
Less than 1 hour	52	15.38	15.38	3.85
1 - 4 hours	664	16.57	13.70	6.63
4 - 8 hours	266	22.93	11.65	7.89
More than 8 hours	51	17.65	17.65	13.73

Table 4.23: User feedback with daily phone use.

Answer to Sub-RQ5. Statistically significant differences were reported in the amount of written feedback given based on the type of software used and the duration of daily use. Respondents who spend more hours each day on their computer reported giving more written feedback to product forums. Those using the Linux OS gave more written feedback to app stores and product forums than those using Windows and Mac. Android users reported giving more written feedback to app stores than iPhone users.

4.1.4 Discussion

Implication 1. *The findings presented in Study IV suggest that to get the most representative user views and desires, feedback from multiple feedback channels should be considered when leveraging online user feedback.* I found statistically significant differences in the users who reported giving feedback on app stores, product forums, and social media concerning traditional demographics and software usage habits. For example, older respondents prefer product forums to app stores, while younger respondents prefer app stores.

Notably, a majority of feedback-giving respondents reported only engaging with one of these three feedback channels. This indicates that considering multiple channels will enable feedback from a more diverse set of users.

I also found key differences in what motivates software users to engage with each of the three channels. The most cited motivation on app stores and social media was to show appreciation for the app/software. Whereas, on product forums showing appreciation was much less of a motivating factor. Instead, getting help was the top-cited motivation.

Showing dissatisfaction, recommending, and discouraging others were also significantly more cited on app stores and social media. On social media, connecting with other users was reported to be a more common motivation than on the other channels.

These motivation differences suggest that the feedback on each online channel is likely to contain different product development insights. For example, feedback on product forums contains users trying to get help and, therefore, likely describes how the software is unintuitive or challenging to use. On app stores and social media, users are more motivated to communicate how they feel about the software/app to the developers and other users. These differences emphasize the benefit of considering feedback from all channels, as each channel may provide unique insights.

Implication 2. *My findings suggest possible approaches to encourage feedback from underrepresented groups and new directions for investigation.* I saw that some demographics were less likely to give feedback than others. For example, respondents 35-44 years old report to provide the most feedback on all three feedback channels, while both older and younger respondents gave less feedback (Figure 4.4). Also, men reported giving feedback at a higher rate than women across all three channels. This is in line with the results of Guzman and Rojas (2019), who found that the Apple app store had more feedback from men.

Underrepresented groups cited several reasons not to give feedback. Women more frequently (than men) reported that they found app stores confusing or hard to use, felt a resolution would take too long, and to not be aware feedback could influence software improvements (Table 4.14). More under 25's than older respondents found app stores confusing or hard to use, reported to not be aware feedback could influence improvements, and felt a resolution to feedback would take too long (Table 4.15).

Therefore, to encourage feedback from underrepresented groups, I propose that the above reasons not to give feedback should be addressed. Online feedback channels should make their interfaces easy to use, for these groups, and add clear messaging about the potential to improve the software. Methods proposed by my respondents may help address these issues. For example, giving a quick response to feedback could emphasize the connection to software improvement and help address the perception that a resolution will take too long. Clearly showing a track record of addressing feedback would also promote awareness of the process and help motivate user input.

Recent research found that most software has gender inclusivity issues (Burnett et al., 2016), so similar inclusivity issues may exist in the software that collects online feedback. Future work could investigate feedback interfaces that underrepresented groups find encouraging and easy to use. Lab trials could be carried out to evaluate if the approaches identified above encourage feedback in a practical context. The option to give feedback through a smart assistant, which was more commonly endorsed by women (than men) (Table 4.17), could be included in the evaluation. Additionally, new chatbot-based approaches are well suited to a lab evaluation of the impact on motivating feedback (e.g., by implementing Ladderbot as a tool to encourage feedback), as I proposed in Studies I and II.

Finally, understanding the coverage of requirements extracted from feedback has practical implications for requirements elicitation practices. Overall, I found that women, users younger than 35 or over 44, and those who use software less are somewhat underrepresented in software feedback. I imagine multiple approaches to be promising and vital to improve the quality of requirements sourced from online feedback. Firstly, tracking meta-information when specifying requirements (such as age, gender, usage experience, usage goals, usage preferences) can help to make requirements coverage more transparent. RE practitioners can use such meta-information to explore whether requirements are evenly representing relevant user groups. Specifically, linking requirements such as user stories to underlying personas could help to document meta-information (De Oliveira et al., 2020; Ferreira et al., 2018). Secondly, visualizing a hierarchy of requirements based on the linked meta-information and thus underlying user groups could help with identifying "blank spots" of requirements coverage. Visualizations could include the relevant target markets and expected or targeted user groups for a software product and show how requirements link to user demands or feedback (Reddivari et al., 2012; Stanik & Maalej, 2019). Thereby, practitioners could spot user groups that are inadequately targeted with requirements. Further, a clear hierarchy of requirements could support adequate requirements prioritization focusing on a broad user inclusion. Thirdly, organizations would benefit from including methods to collect and analyze information from underrepresented or missing user groups into elicitation processes to fill blank spots. Therefore, interviews or focus groups could focus specifically on underrepresented groups. Modern solutions could be applied to extend feedback and requirements elicitation to a wide audience of users, including a sample different from regular online feedback channels. As demonstrated in Study II, chatbots show promise for eliciting requirements at scale.

Implication 3. *Feedback prompts effectively elicit feedback for app stores and may be effective if applied more widely in computer software. However, many respondents reported being annoyed by prompts and rushing to close them. This is likely a factor in prompts not being as effective at eliciting detailed feedback.* Mobile apps widely use prompts to elicit feedback. More survey respondents reported giving written feedback on app stores than on any other channel. Much of this feedback is prompted. The number of respondents who have provided unprompted app store feedback (12.39%) is very similar to the number who report having written posts on product forums (13.45%). This suggests that the prompts are successful in eliciting additional feedback givers. The prompts are even more effective at eliciting app ratings, which take less time to provide than written feedback.

However, many respondents reported being annoyed by prompts in their open-ended responses. One respondent said *"The number of times they asked me to rate it was getting annoyingly high, so I just did it so they would stop prompting me".* This may partly explain why prompts are not as effective at eliciting written feedback. Users want to get rid of them and often just give a quick rating. There may also be a danger that prompts negatively affect user experience.

Future research could investigate if prompt timing and frequency affect the likelihood of eliciting written feedback and their effect on user experience. Other prompt types, such as multi-choice questions, could be trialed in place of open-response fields to elicit detailed feedback. Additionally, ways to integrated prompts into other feedback channels can be investigated.

Implication 4. *The types of software devices respondents use also have an association with feedback habits. Investigating why users of some devices give more feedback may give insights into how to motivate and facilitate feedback.*

On phones, more Android users give written feedback than iPhone users. It is not clear why there are differences in feedback across devices, but it may be influenced by differences in prompt rates, app quality, app store usability, or even those who choose to use each phone type. iOS developers could benefit from understanding these factors in order to encourage more feedback from their users.

On computers, respondents who use the Linux OS more commonly had given written feedback to app stores and product forums than those who do not use Linux. The feedback habits of Mac and Windows users were relatively similar across all feedback channels. The higher feedback rates of Linux users may be related to the prevalence of software developers using it. In fact, 43% of respondents using Linux also reported working in the software industry, compared to only 16% of all respondents. My results showed that software professionals are more likely to provide online feedback, possibly because they understand how development teams will use that feedback. Future research can investigate more thoroughly the reasons for differences across devices.

Implication 5. *My findings suggest approaches to motivate user feedback, which can be employed when more feedback is needed, such as for new applications or those with small user-bases.* User feedback serves two primary functions, it is used as a source of requirements by developers, and potential users consider reviews when choosing applications (Pagano & Maalej, 2013). A lack of user feedback can limit new applications and those with small users bases. Previous research has even highlighted the issue of small applications paying for ”fake” reviews (Martens & Maalej, 2019).

This study suggests approaches developers can use to elicit additional feedback. A small financial incentive was the most commonly endorsed method to encourage user feedback (82%) (Figure 4.6). In-app rewards were the second most popular potential encouragement method (65%) and maybe a more realistic option for apps with limited resources. Previous research found that in-app rewards such as digital goods, game progression, and customization options can motivate user behaviour(Bleize & Antheunis, 2019). The ability to give anonymous feedback could also encourage additional user engagement, as was suggested by multiple survey respondents independently. However, the benefits of anonymous feedback must be weighed against the possibility of encouraging more fake reviews and reduced user accountability for the quality of their feedback (Martens & Maalej, 2019). Future work could look at ways to satisfy (some) user’s desire for anonymity while still maintaining user accountability. One approach could be to allow feedback through existing accounts, such as Google or Facebook, and not accessing or sharing account details while the terms of services are adhered to.

My findings also suggest that providing quick responses to feedback givers can encourage feedback and show a clear track record of addressing previous reviews. Future work could study these approaches' effectiveness when developers use them in practice.

Other avenues for future work. *Investigate other feedback channels.* My study was limited to app stores, product forums, and social media. Future work could perform a similar investigation considering other feedback channels like issue trackers.

Replicate survey in other countries. My survey respondents mainly were from three countries; New Zealand, Germany, and China. Future work could replicate my study by eliciting responses in additional countries. This would also enable analysis at the ethnicity level if more ethnic diversity in the participants was achieved.

Understand gender differences in product forum engagement. In addition to men being more likely than women to post on product forums, men also reported using product forums for different reasons. While men and women both primarily used forums to ask software-related questions, men also reported higher rates of giving other types of feedback on product forums, including: reporting problems, requesting features, praising and criticizing the software, and assisting others. Further research is needed to understand the gender difference in engagement with product forums.

Making missing demographics more transparent. Currently, it is difficult for product development teams to know whether the feedback collected from online feedback channels is biased and misses the voices of some underrepresented groups. Future research could devise ways to make this more transparent to enable software development teams to proactively consider the underrepresented groups' needs and produce more inclusive software.

Investigate differences in feedback rates for different types of software applications. Software users may be more likely to give feedback on some types of software compared to others. Feedback on different software types may also vary between user demographics. Understanding these differences would give valuable context to the requirements sourced from the feedback.

4.1.5 Conclusion

The online user feedback written on app stores, product forums, and social media is a valuable source of requirements for software developers and has been a focus of requirements engineering researchers. However, limited studies have been done to understand which software users give this feedback, what motivates them to give feedback, and dissuades them when they do not. In Study IV, I first directly surveyed 1040 software users about their feedback habits, software use, and demographic information. I then extended the initial results by surveying 936 users on why they do not give feedback when they have software issues and ways to encourage them.

The responses indicate significant differences in the demographics of software users who give feedback on each online channel. For gender, men reported giving more feedback than women, and respondents between 35 and 45 reported giving the most feedback across all channels. I also found strong evidence that younger software users (under 25) prefer to engage with app stores, whereas older software users use product forums at equal (to app stores) and sometimes higher rates.

I identified critical differences in what motivates software users to engage with each of the three channels. Comparing channels, respondents reported the top motivation to give feedback on app stores and social media was to show appreciation, whereas, on forums, the most cited motivation was to get help with software products. Differences between the motivations of men and women to give feedback were also reported for each of the channels. Respondents reported in-app prompts to be significantly more effective in motivating them to give app ratings over written feedback. Additionally, individual feedback givers reported engaging more times a year on product forums than on app stores.

The top three reasons not to give feedback, as reported by respondents, were consistent across the three study channels, if not in the same order, namely: 1) Looking for an existing answer instead, 2) finding an alternative app instead, and 3) feeling a resolution would take too long. Significant differences in the reasons not to give feedback were also identified between men and women and between different age groups. Multiple respondents also reported common additional factors that dissuade them, including wanting to stay anonymous, not wanting to create an account on forums, and not wanting to share software issues on social media.

Respondents saw financial and in-app rewards as better methods to encourage feedback than new options such as giving feedback through a smart assistant. Additional methods to encourage feedback were suggested by respondents in open-ended responses, including: Wanting a quick response to show that feedback had been seen; The ability to give anonymous feedback; And, showing a clear track record of user feedback being addressed.

Differences in feedback habits were also reported with the ways respondents use the software. Those who spend more hours each day on their phone or computer reported giving more feedback about the software they are using. The software platform being used also presented a relationship to feedback rates, with more Linux (computer) and Android (phone) users reporting to give feedback than those who use the alternatives.

The findings presented in Study IV give meaningful insights into which software users give online feedback and the motivations they have to give it. I found notable differences in those who give feedback to each online channel, emphasizing the need to mine all three feedback channels to get the most representative requirements from software users when leveraging online feedback. Reasons software users do not give feedback and methods to encourage them have also been identified. These may give insights into how to improve feedback rates (when they are low), especially from underrepresented demographic groups.

5. Discussion [20]

In this thesis, I explore the design of a requirements elicitation chatbot and an IML system to semi-automate qualitative coding. Further, I present results from a large-scale study on IS feedback engagement. This thesis has several theoretical contributions and practical implications, which I introduce in the following sections. Additionally, I outline limitations and avenues for future work for each study.

5.1 Theoretical Contributions

This thesis contributes with knowledge for designing AI-based qualitative data collection and analysis systems and a deeper understanding of the coverage of existing data collected from online sources. Additionally, I demonstrate the application of a chatbot interviewer to understand user values in smartphones.

In **Study I**, I propose the design and conceptual architecture of Ladderbot, a chatbot capable of facilitating online laddering interviews in a scalable fashion with wide audiences. Therefore, I answer the research question *how a requirements elicitation system could be designed to engage a wide audience of users, regardless of previous experiences with contributing requirements in IS development projects?* (**RQ1**). I initially aggregate the prevalent challenges that RE interviews with novices face from existing RE and HCI literature: (1) the need for a fixed structure for elicitation interviews, (2) a lack of interview depth, due to not enough *why?* questions being asked, (3) ambiguous statements at an insufficient level of abstraction, (4) lacking help for novices with visualizing relationships, and (5) a lack of technical and soft interviewer skills with regards to question formulation, ordering, omission and behavior. In a second step, I identify laddering as a promising approach to interviewing novice users due to its benefits for clarifying requirements, inherent hierarchical nature, and effectiveness for eliciting information. Laddering interviews, on the other hand, face several issues. Laddering can be a monotonous and tiring interview technique, requires highly trained interviewers, and has a risk of interviewer bias affecting the interview. The benefits of chatbots, such as an effortless, barrier-free interaction and dialogue guidance, provide ideal grounds for circumventing the issues of laddering interviews – making the combination of chatbots as technological and laddering interviews as methodological foundation highly promising as design for an RE chatbot. Related tools that allow users to communicate requirements do not consider users' experience level, limiting the utility for novices (Kato et al., 2001; Mohedas et al., 2015). By proposing a design for a laddering interview chatbot that does not require prior domain knowledge for its configuration, the bot can conduct exploratory interviews with novice users. Commonly, chatbots require domain-specific training data to ask questions and identify intends during an interview (e.g., Rajender Kumar Surana et al. (2019)). These training requirements make the tool a bad fit for cases where no domain knowledge is available, such as new software development projects or extensions to new domains. With Study I, I take an

[20]This chapter is based on the following studies which are published or in work: Rietz and Maedche (2019), Rietz and Maedche (2021b), Rietz and Maedche (2021a), Tizard, Rietz, Liu, et al. (2021).

essential step towards creating effective RE chatbots, which has been lacking from the existing body of knowledge (Dieste & Juristo, 2011) by providing design knowledge for laddering interview chatbots capable of conducting exploratory interviews with novice users.

In **Study II**, I evaluate the chatbot design outlined in Study I by comparing the laddering chatbot against two forms of established survey-based approaches for online laddering interviews. Thereby, I analyze *how the results of chatbot-based laddering interviews compare to established survey-based laddering approaches* (**RQ2a**). I conduct laddering interviews with 256 participants in three treatments on user values in smartphone use. This case recreates an experiment conducted by Jung (2014), thus providing a realistic scenario for evaluating descriptive and qualitative differences between the treatments and judging the quality of the collected data. The findings for this case allow me to grasp *what insights chatbot-based laddering interviews provide to help us understand user values in smartphones* (**RQ2b**). Therefore, I coded the interviews to understand users' hierarchical goal structure of smartphone usage behavior. I find that (1) smartphones are a means for users predominantly to communicate and achieve socialization, (2) users pursue intellectual and emotional self-optimization to achieve satisfaction, and (3) users prioritize social and utilitarian values over achieving convenience. Additionally, I investigate negative gains of smartphone use and find that users are wary of how smartphones promote and force behavioral change. Overall, survey-based laddering more reliably produces ladders that end in values, while chatbot-based laddering sacrifices clear attribute-consequence-value structures to explore negative gains. However, the chatbot engages participants to give significantly more and longer answers and guides participants during the interview process, resulting in significantly higher learnability. My findings from Study II have twofold theoretical implications. **Firstly**, my study demonstrated how smartphones and their integration into everyday life have changed since 2014 and presents a bottom-up view of smartphone acceptance and values based on both positive and negative gains. Interestingly, the value *confidence*, being the most central value in Jung (2014) 's original study, was discarded as a value in my study. Participants' answers did not fit Jung's definition of confidence as a feeling of superiority towards others, and answers instead reflected the values *self-optimization* and *satisfaction*. I find that using smartphones for communication is crucial for participants to achieve socialization and kinship. Notably, users might be willing to sacrifice some convenience to improve their ability to achieve these primary ends. Furthermore, I highlight the importance of *self-optimization* for smartphone use, with some users using the technology as a natural extension of the own self. I also looked into the negative gains of smartphone use. The bottom-up view of laddering provides critical insights into how and why users interact with capability-augmenting technologies and why new capabilities might force behavioral changes. Therein, I contribute knowledge to understand hierarchical goal structures of user values and negative gins in smartphone use. **Secondly**, I demonstrate the benefits and challenges of chatbot-based laddering. While participants provided more and longer answers to the chatbot and reported a more straightforward interview process, the resulting semi-soft interview structure complicated the analysis process. Further, not all projects may have easy access to chatbots that can be configured according to the design outlined in Study I. To that end, I demonstrate that survey-based laddering also produces commendable and well-structured results by comparing survey- and chatbot-based laddering. Visualizing the laddering structures during survey-based laddering did not have a significant impact compared to regular PP laddering. Overall, my findings suggest combining manual and chatbot-based interviews to engage users in wide audience laddering studies. Study II extends the body of knowledge by presenting significant benefits of laddering chatbots over laddering surveys as well as demonstrating important drawbacks. Further, I contribute to value-oriented research by showing the positive and negative gains achieved with smartphone use with the study.

In **Study III**, I propose a system to support semi-automated qualitative coding and investigate how qualitative researchers use the system to code a dataset while receiving code recommendations. Thereby, I illustrate *how to design an IML system to support qualitative coding* (**RQ3a**) and demonstrate *how qualitative researchers use an IML system compared to the commercial and well-established QDAS MAXQDA* (**RQ3b**). In Study III, I aggregate related work on coding in QDA, available QDAS, and AI-based qualitative coding. As I identified a literature gap regarding the integration of rule definition and ML model training into the qualitative coding process while providing trustworthy suggestions, I define six requirements for building an IML system that emphasizes an interactive AI-supported coding process. In short, the system needs to enable users to (re)define code rules while automatically refining an underlying ML model to support coders with suggestions iteratively. These suggestions should foster reflection and include explanations. I conduct both a formative and a summative evaluation to understand how researchers interact with my prototype system Cody. While I use the formative evaluation to refine Cody, the summative evaluation contributes to understanding how researchers interact with an IML system that assists users during coding. I find that (1) code rules provide structure and transparency, (2) explanations are commonly desired but rarely used, and (3) suggestions benefit coding quality rather than coding speed, increasing the intercoder reliability, calculated with Krippendorf's Alpha, from 0.085 (MAXQDA) to 0.33 (Cody). Further, with Study III, I contribute a concept for automatically creating code rules suggestions, which users value as a support for defining and revising their code rules. Additionally, I present a strategy for making ML-based code suggestions with a low number of examples by defining strict quality criteria and including artificial negative examples in the model's training. The design implications I present include suggestions on integrating code rules and rules suggestions without compromising researchers' agency. Suggesting one specific label for a text instance before a user has interacted with the respective instance can tempt the user to accept the suggestion to reduce the workload. I present strategies for reducing these risks. Specifically, systems could willingly reduce the precision of suggestions by suggesting multiple codes instead of one, or suggest codes only after an annotation is made, rather than preemptively annotating sections in a text. With Study III, I contribute design knowledge for semi-supervised qualitative coding systems. Further, I contribute descriptive knowledge to understanding users' interaction with IML coding systems by demonstrating the benefits and challenges of code suggestions and suggesting explanations.

In **Study IV**, I shift the focus from investigating tools for data collection and analysis to learning more about the users themselves. Therefore, I conduct two online surveys to investigate *what the demographics of software users that provide online feedback are, and what motivates their feedback behavior* (**RQ4a**). In an initial survey of 1040 software users, I identify statistically significant differences in the demographics of users who give feedback (gender, age, etc.) and key differences in what motivates them to engage with each of the three studied channels (app stores, product forums, and social media). In the second survey of 936 software users, I identify *what motivates software users to give online feedback today, and what could enhance the motivation to give feedback in the future* (**RQ4b**), including significant differences between demographic groups. I also present a detailed list of user-rated methods to encourage their feedback. Specifically, I find that (1) traditional demographics, such as gender or age, significantly influence software users' feedback behavior, (2) the motivation to provide feedback varies across the three feedback channels, and (3) while users report being most interested in incentive-based elicitation methods, participants agree that alternative elicitation options could encourage feedback. Study IV provides a meaningful context for requirements sourced from online feedback, identifying underrepresented demographic groups. Findings on what motivates and discourages user feedback give insight on how feedback channels and developers can increase engagement with their user base. With Study IV, I contribute descriptive knowledge to understand who provides feedback in online channels today and outline why. Further, I provide suggestions for improving feedback rates in the future, especially from underrepresented demographic groups.

To summarize, in part I of this thesis, I contribute with the design of Ladderbot, a tool for collecting laddering interview data, and Cody, a tool that provides users with IML support to analyze qualitative data. In three studies, I provided detailed design recommendations, present the artifacts, and outline strategies for evaluating both artifacts. In Study I, I propose a design and architecture for Ladderbot, a chatbot that tackles common issues with RE interviews with novices. In Study II, I present a case study of chatbot laddering interviews for eliciting user values in smartphone use. In Study III, I demonstrate the utility of an IML-based QDAS compared to an established QDAS as a baseline. Thus, part I contributes to the understanding of 1) how users interact with chatbot interviewers compared to surveys in the case of laddering, and 2) how users interact with IML support for data analysis, focusing on criteria such as coding speed, coding quality, trust, and researcher agency. In part II, I step away from designing and evaluating artifacts to survey almost 2.000 software users directly. With Study IV, I contribute to understanding the demographics and motivations of software users for providing feedback. Additionally, part II prompts users for their perception of new approaches for collecting feedback. Throughout this thesis, I present the top-down design (based on gaps in the body of knowledge) and evaluation of tools, as well as a bottom-up exploration of feedback behavior. Thus, this thesis covers multiple steps of the elicitation process, most importantly, data collection and analysis[21]. Rather than focusing on one step, the approach of my thesis is to investigate both the

[21] I also investigated other steps in the requirements engineering process, such as requirements prioritization (see Rietz and Schneider (2020)), which are not included in this thesis

	Study	Main Theoretical Contributions
Part I	I	- Design knowledge for a laddering interview chatbot that does not require prior domain knowledge for its configuration and can conduct exploratory interviews with novice users
	II	- Provision of a hierarchical goal structure of user values and negative gains in smartphone use - Understanding of how users interact with chatbots interviewers compared to surveys
	III	- Prescriptive knowledge for designing systems for semi-automated qualitative coding - Understanding of how users interact with IML for data analysis support and how it impacts coding quality, speed, and intercoder reliability - Concepts for providing explanations and code suggestions to minimize the impact on researchers' agency
Part II	IV	- Descriptive knowledge of the impacts of demographic factors on feedback behavior - Insights into the motivations of user groups for giving feedback and the top reasons why users do not give online feedback - Overview of promising methods to encourage online feedback
Overall		- Design implications for building laddering interview chatbots and IML-based QDAS - Descriptive knowledge of user engagement with interview chatbots compared to surveys, and with IML-based QDAS compared to established QDAS - Hierarchical goal structure of positive and negative gains of smartphone use - Improved understanding of demographics and motivations of software users for providing feedback, with regards to common and novel approaches to encourage feedback

Table 5.1: Theoretical contributions of this thesis.

challenges of data collection with chatbots and what to do once large amounts of data have been collected. This thesis sheds light on an end-to-end process by providing an overview of challenges and presenting and evaluating possible solutions along the entire process of wide audience user involvement. Specifically, my findings highlight the benefits of a laddering chatbot compared to surveys and show that surveys can also provide commendable and critical results. The extensive hierarchical goal structure that I created by combining the two data collection methods highlights the utility of both approaches. For analyzing the resulting (laddering) data, I provide a tool that can substantially support the analysis process. However, I set out to accelerate qualitative coding yet found that Cody has a more substantial effect on coding quality than coding speed. While improvements are required to achieve an acceleration of the coding process, which I outline as part of Study III, my findings contribute knowledge to understanding how users interact with IML-based support during coding. I also demonstrate the discrepancies in explanations that users desire and how they are used when available. As the studies I-III followed a top-down approach to creating artifacts based on the related work and the available body of knowledge, Study IV completes this thesis with a bottom-up approach by probing users for their motivations to contribute feedback. Interestingly, users desire financial incentives and rewards for their feedback, while using smart assistants to encourage feedback, e.g., chatbots, was not favored. These findings open up relevant opportunities for future research. In Study II and related work (e.g., S. Kim, Lee, et al. (2019) and Tallyn et al. (2018)), I see evidence that users enjoy and interact with chatbots more than with traditional surveys. Thus, future studies should explore these discrepancies between users' expectations of chatbot interviewers and perceptions during or after their use. These results may also be influenced by the gender of the interviewee, as indicated by Study IV. Overall, this thesis provides insights into and extends the understanding of using AI-based qualitative data collection and analysis systems and the coverage of already existing data collected from universally accessible sources. The main theoretical contributions of the studies in this thesis are summarized in Table 5.1. My findings contribute to research in the domains of RE, IS, and HCI.

5.2 Practical Implications

On the practical side, this thesis contributes tangible artifacts, architectures, evaluations, and guidelines for multiple stakeholder groups.

In **Study I**, I present a domain-agnostic architecture for laddering chatbots. Chatbots require upfront training data to enable the bot to identify key utterances during conversations and react accordingly. However, such training data can be hard to acquire for practitioners, especially when engaging in exploratory studies (e.g., developing software in a new scenario or novel domain). The architecture I suggest for Ladderbot uses the generic laddering interview structure to guide interviews without requiring upfront training data. Therefore, it utilizes default *why* questions, alongside four randomly selected additional prompts. Researchers and practitioners alike can benefit from the proposed structure for implementing their laddering interview chatbots. Additionally, I contribute Ladderbot as a freely available artifact with a creative commons license. The artifact can be easily adjusted for different study domains by adjusting the 29 integrated questions. Furthermore, Ladderbot includes preset weights that can be used to manipulate the questioning structure.

In **Study II**, I outline the hierarchical goal structure of positive and negative gains of smartphone use. Thereby, I offer insights for players in the IT sector, e.g., app developers, hardware providers, or communication companies. My findings extend prior work by giving a recent bottom-up view into both positive and negative value-oriented achievements of smartphone use, collected from a wide audience of smartphone users in Germany. Firstly, I present possible focus areas for smartphone marketing campaigns that allow the targeting of key values in smartphone use. Specifically, marketing campaigns should address communication and socialization when focusing on social values or target productivity improvements or entertainment benefits when focusing on utilitarian values. Secondly, I outline approaches for improving work efficiency through smartphones while circumventing typical negative gains. Organizational commitment is vital to drive the utilization of phones for professional use, e.g., by providing app-based solutions for organizational tasks. However, companies must be wary of supporting the employees with being in charge of their time, protecting them from perceiving work phones as a sole source of stress. Thirdly, my findings imply that players in the IT sector may achieve differentiation in the commoditized app market by designing offerings around users' most central values: self-optimization, socialization, or satisfaction. Additionally, companies should explore user concerns, e.g., impersonal communication or time waste, to provide offerings that support a conscious, attention-aware use of apps and devices.

In **Study III**, I present an artifact supporting qualitative coding in a semi-automated fashion. While the artifact provides a tangible contribution that may help with qualitative data analysis, a more vital contribution for practitioners may be found in the outlined system requirements and design recommendations. I present six requirements for building assistive tools for qualitative coding, focusing on relevant pillars for ML-based coding support, which I derived from an extensive study of relevant literature on (AI-based) qualitative coding. Furthermore, I present possible solutions for suggesting code rules to users and for training supervised ML models with sparse training data. Specifically, I propose using a combination of a defined code and the respective section in the data for creating a possible code rule to suggest to users. While a code rule created with such a simple rule likely will not be perfect, my evaluation highlighted that code rules suggestions are invaluable for users to understand their purpose and structure. Further, imperfect code rules may encourage users to make (iterative) changes. For model training, I propose utilizing skipped sections in the coding process as examples for sections that might not interest the coder. While this approach has limitations, which I outline in Section 5.3, the strategy may inspire practitioners to use implicit information (e.g., coding behavior) to provide an ML model with more information during training.

In **Study IV**, I provide insights into the demographics and various motivations of software users for giving feedback online. The findings can be used in two ways: firstly, for coming up with requirements for better and more evenly gathering feedback for software. Specifically, online feedback channels should make their interfaces easy to use, particularly for groups whose feedback is underrepresented. Further, smart assistants can help replace feedback prompts, particularly for eliciting written feedback. Similarly, assistants could help with outlining the impact of feedback to users and help with showing the results of the provided feedback. Secondly, the findings can help understand the coverage of requirements extracted from feedback. Overall, there is more feedback from males, more feedback from users with software experience, or users who use software longer. This imbalance of requirements coverage has implications for RE practitioners, for example, for balancing software feedback from online channels with other sources of elicitation (e.g., interviews or focus groups). Tracking meta-information for requirements, such as the linking requirements to respective users or personas, could allow practitioners to judge the distribution of requirements regarding the target user groups. Criteria for an even distribution could be age, usage preferences, technology experience. Likewise, visualizing a

	Study	Main Practical Implications
Part I	I	- Domain-agnostic architecture for developing a laddering chatbot - Artifact for conducting laddering interviews with wide audiences
	II	- Focus areas and insights for players in the IT sector, such as app developers, hardware providers or communication companies to inform marketing campaigns and app development - Suggestions for organizations to encourage using smartphones to support work practices
	III	- Recommendations for how to generate code rules suggestions and for training a supervised ML model with sparse training data - Artifact for supporting qualitative coding in a semi-automated fashion with code rules and ML-based suggestions
Part II	IV	- Guidance on how to improve working with feedback from app stores, social media, and forums - Guide to understanding the role of feedback in the broader chain of requirements elicitation - Insights to develop methods for better and more consistent collection of feedback for software
Overall		- Demonstration of benefits and guidance for designing chatbots as laddering interviewers and IML systems to support qualitative data analysis - Suggestions for players in the IT sector to inform smartphone-related offerings and utilize smartphones as a productivity tool - Guidance for improving the requirements coverage from software user feedback

Table 5.2: Practical implications of this thesis.

hierarchy of requirements based on an underlying user group could help identify if groups are adequately included and whether requirements represent target markets and user groups. Additionally, hierarchies could help spot differences between groups and prioritize requirements adequately, focusing on a broad user inclusion. Overall, the identification of scarcities in requirements coverage is only the first step. Companies subsequently require ways to collect and analyze information from underrepresented or missing groups, some of which I proposed and evaluated throughout this thesis.

To summarize, this thesis has implications for three primary, yet not necessarily exclusive, stakeholder groups: *qualitative researchers*, *requirements engineers*, and *players in the IT sector*.

Firstly, qualitative researchers and requirements engineers may utilize my artifacts to collect and analyze data. So far, I made Ladderbot available as an artifact under a creative commons license (see Rietz and Maedche (2019)). Cody is available online upon request. More importantly, practitioners may find the architectures, parameters, procedures, and technology stacks outlined in this thesis valuable for implementing their instantiations of interview chatbots and IML-based QDAS. Throughout studies I-III, I present concepts for translating the findings and theoretical contributions to potential system improvements. To give two examples: firstly, I outline potential improvements to the interview structure of Ladderbot by dynamically adjusting the weights of questioning techniques based on the interviewee. Secondly, I discuss strategies for enabling users to work with code rules through rule suggestions and utilizing "ignored" sections of a document for ML-model training.

Requirements engineers, in particular, can use the findings and suggestions in Study IV to improve their elicitation processes. By considering the demographic coverage of readily available data in common feedback channels, practitioners can make adjustments to the applied complementary elicitation methods (e.g., interviews, focus groups) to focus specifically on underrepresented groups. Additionally, I suggest strategies for enhancing the feedback behavior of users, which may be valuable for supporting users in giving feedback beyond app stores and social media.

Finally, players in the IT sector may find the strategies deployed in Study II insightful to engage with users and continuously monitor developments in usage behavior and values associated with smartphone use. The hierarchical goal structure of smartphone use may help inspire and inform marketing and product evolution strategies. It may also inspire a regularly updated user involvement process, potentially based on the collection and analysis artifacts presented in this thesis. Additionally, the findings from the real use-case of collecting data with Ladderbot may help IT sector players and companies in other domains alike with improving work efficiency through smartphones. Overall, this thesis provides artifacts, architectures, and solutions for implementing AI-based qualitative data collection and analysis systems, guidance for extending the coverage of feedback from software users to underrepresented user groups, and suggestions for utilizing the prevalent user values achieved through smartphone use. The main practical implications of the studies in this thesis are summarized in table 5.2.

5.3 Limitations and Future Work

All four studies in this thesis were conducted with an emphasis on rigor and relevance. However, some limitations remain and should be addressed in future research. In the following, I outline the limitations and implications for future work of each study.

Study I

Study I focuses on design knowledge for a laddering chatbot. In the following, I rely on some of the insights and feedback collected from users in Study II to present these limitations better. Firstly, students might be keener to commit to an automated interview than non-student users, especially when facing issues during the interview process, as one user stated: *"The bot did not build upon my answer two times, which is okay for a chatbot"*. Students might be more forgiving to errors in an interview than, for example, practitioners, especially when incentivized in an experimental context.

Further, some issues arise from the combination of a chatbot and the laddering technique. Participants interacting with the chatbot are looking for a "human-like" interviewing experience. As an indicator, participants seemed to pay special attention to a wide range of questions and careful incorporation of previous answers. However, the laddering technique utilizes a quite monotonous questioning strategy, constantly focusing on uncovering the *why*. Hence, some participants believed that the chatbot failed to understand them and kept asking repetitive questions. One user reported: *"You feel as if talking to a machine that keeps asking the same questions, or if talking to a child that keeps asking why?"*. Future research should evaluate ways of making laddering interviews with laddering chatbots feel more dynamic, potentially by adapting the weights of the decision gates based on provided answers. Furthermore, as some participants reported that they lacked social cues during the conversation, subsequent iterations of Ladderbot should include social cues to make the conversation appear more human-like, e.g., by varying response times or by expressing content from earlier parts of the conversation (Gnewuch, Morana, Adam, et al., 2018). For future iterations of Ladderbot, I envision allowing users to edit answers and jump between attributes by using the interview visualization as a navigation device. Users could add information to existing ladders, which would allow Ladderbot to perform soft laddering as well. Thereby, users may experience more dynamic, interactive laddering interviews, allowing them to add and extend the information they did not think of at an earlier stage.

Study II

Study II focused on available technology-enabled techniques for wide audience laddering interviews. While VPP featured an extension of established laddering surveys, PP used the standard procedure and questions. Therefore, I collected data with a baseline method as a benchmark for chatbot laddering. However, I did not conduct any manual interviews with participants as a benchmark for chatbot- and survey-based laddering. Manual (face-to-face) interviews could help to understand my results better, and particularly probe for the most consequential (e.g., the prioritization of self-optimization and socialization over convenience) or the rarest ladders (e.g., achieving autonomy through smartphones).

Further, Study II was conducted entirely with European university students, primarily based in Germany. This sampling may limit the generalizability of the outlined findings. However, related work on smartphone values worked with student samples too, often in South Korea (Chun et al., 2012; Jung, 2014; K. M. Kim & Hwang, 2020; C. Y. Lin et al., 2017; J. Park & Han, 2013). Comparing my results to other student samples may benefit the validity of the findings. Compared to South Korea, Germany is in the centerfield regarding mobile infrastructure and services (OpenSignal, 2016), potentially making the results more representative for the general (student) population in western cultures. Another limitation for Study II is the interaction design used by the interview chatbot. Users had to switch between ladders and end the final ladder with the *stop* command. As such, finishing an interview with the chatbot resembles an *opt-out* procedure while continuing the interview with an online laddering survey resembles an *opt-in*. It is known from organ donation that changing the default option from opt-in to opt-out can significantly increase donation rates (Ahmad et al., 2019). My findings regarding the stop rate support this explanation, as more than half of the participants in LB completed the entire interview without using *stop* once (54.12%). In comparison, less than 3% of the participants in PP and VPP answered every possible question in their interviews. While this argumentation may provide one explanation as to why participants in the chatbot treatment provided significantly more answers, the presented stop-rate might be offset by some users having overread the instructions regarding how to use the stop command. However, I provided this information on multiple pages throughout the introduction and during the interview.

Besides the interaction design, a potential weakness lies in the interview structure of the chatbot. *Firstly*, the chatbot used only rudimentary strategies to react to interviewees, based on human-defined reactions and rules. Hence, in some conversations, the chatbot did not respond appropriately. I removed apparent interview "failures" during the data preprocessing. Furthermore, I did not observe significant differences in participant enjoyment between my treatments. Still, a more sophisticated question structure and methods for reacting to answers and asking follow-up questions can make the interaction more detailed and fruitful. As such, I am excited to see future chatbots provide a significantly improved interview experience. *Secondly*, chatbot's prompts for negative gains disrupted the means-end question structure by introducing a semi-soft laddering structure. As a consequence, fewer ladders ended in values, compared to survey-based laddering. Future designs should improve upon integrating negative gains into laddering interviews or targeting either positive or negative gains.

Another potential limitation to Study II results from some values or linkages not being considered due to their rare occurrence below the cutoff value. With more participants, these nodes may become relevant. With access to techniques for wide audience interviewing, researchers need to reconsider the appropriate size of a sample to reach theoretical saturation. For laddering studies, Reynolds and Gutman (1988) consider a pool of 50-60 informants to be appropriate to address a research question, while Reynolds, Dethloff, et al. (2001) suggest 20 well specified and screened participants as a rule of thumb. With techniques that allow the involvement of a magnitude of participants, future studies need to guide how to approach sampling participants in wide audience interviews.

Besides providing guidance on approaching data collection from a wide audience, there is value in supporting the data analysis step for qualitative research studies with large sample sizes. In Study II, I analyzed data from 256 interviews, which was a time-consuming and repetitive task. Future research should evaluate the quality and access implications of using interactive AI-based systems to semi-automate the coding process (Marathe & Toyama, 2018; Rietz & Maedche, 2021a). In laddering, tool support can be valuable both for content coding and for the generation of AIMs and HVMs from coded data. The automatic generation of HVM drafts at various cutoffs can support researchers with immersing in their data and comparing treatments. Furthermore, automated HVMs can support the identification of errors in figures that were manually crafted.

Finally, Study II is limited by the methodological design not encompassing the societal impacts of smartphone usage. Effects of smartphone usage (e.g., changes in communication styles and impact on work-life balance) have transformed and will continue to transform culture and social structures. Future research is required to investigate the social values of smartphones and the effects of personal goals of smartphone usage on society. Finally, my study does not use remote interview technologies to include participants from multiple demographics. There is ample opportunity to conduct value-oriented studies with automated interviews with varying sample demographics (e.g., age, ethnicity, education, location). Interview assistants enable participants to participate at their own schedule, potentially allowing previously underrepresented groups to participate in studies. Large-scale studies with diverse users can help to understand (end-user) technologies on a global scale.

Study III

For Study III, I see several ways to improve upon the study and extend the findings. *First*, participants in the summative evaluation worked with data they had never seen before. Additionally, I told them that coding would take approximately 8 hours. Therefore, the evaluation results regarding coding time can serve only as an indication of the effects of interacting with suggestions. *Secondly*, participants did not use their coding after the experiment, giving them little incentive to code to the best of their ability. However, I ensured that they did not know whether they would be asked questions about the content or their coding during the final face-to-face interview. The interviews used for coding had roughly 18.600 words, which some participants perceived as too little to make use of automation appropriately. It would be interesting to test Cody *in the field* with the researchers' projects, where researchers deal with more data without an estimation of how long coding will take. A field evaluation can also help address other limitations Study III: I explicitly encouraged participants to add their codes to the provided codebook if necessary. While instructions and codebook provided participants with a common coding goal, it may have restricted participants in applying their coding style. Further, the presented results on intercoder reliability are illustrative only for the codebook research method. A field evaluation could evaluate Cody's impact on other kinds of qualitative research – I expect that the utility of code suggestions might shift towards assisting in uncovering ideas and themes during codebook development. For some coding strategies (e.g., in-vivo coding), the utility of code suggestions may be limited. *Thirdly*, my strategy to creating artificial negatives assumes that users code linearly from top to bottom and rarely miss important sections during coding. Further, when using rule suggestions for model training, imprecise or wrong rules can cause errors to propagate, resulting in wrong ML suggestions. In the end, the amount of available training data limits the quality of ML-based suggestions. Participants with Cody made, on average, 182 annotations for 38 labels, resulting in a very spare training set. While I improved the ML model(s) through greygoo labels and one-versus-rest training, the quality of ML-based suggestions during the evaluation was limited. However, my aim was not to improve model training in a cold start case but to understand how participants interacted with ML suggestions. The results of Study III indicate that with artificial negatives, learning from rule suggestions, and careful filtering, ML-based suggestions can be used even in a cold start case with sparse training data. An avenue for future work is to evaluate different data collection strategies for cold-start model training. Integrating other technologies to recognize sections that coders intentionally did not annotate, such as eye-tracking, could be an exciting research opportunity (Toreini et al., 2020). Further, participants coded the same documents predominantly using the same codebook, yet Cody trained the ML model individually for each user. Training a shared model on the examples from multiple coders could increase the quality of ML-based suggestions. Finally, Study III focused on each coder working on an individual copy of the data. Integrating and evaluating mechanics for multiple coders to collaborate in coding documents could extend this work. It would be interesting to observe whether formulating rules can help multiple coders discuss their interpretation of labels and how coders work with suggestions based on their co-coder's code rules.

Study IV

Firstly, Study IV is limited by the convenience sampling strategy that I used in both surveys to elicit survey participants, which is a non-probabilistic sampling method and a possible source of bias (Etikan, 2016). The target population of the studies are users of software and mobile applications. In the first survey, participants were engaged via Facebook, Twitter, the hroot participant pool of the Karlsruhe Institute of Technology (KIT), and in Auckland cities public areas. In the second survey, participants were engaged through the hroot participant pool, Zhejiang University's online student forums, and a follow-up email to the first survey participants. Therefore, in both surveys, only a subset of the target population had the opportunity to participate. Additionally, all respondents who completed the survey were self-selected, and their feedback habits may not generalize to all software users.

To mitigate this bias, I collected data from a large number of software users, 1040 participants in survey one and 936 participants in survey two. Recruitment was done through multiple channels to increase the chances of recruiting a diverse set of respondents. However, I cannot claim that my results generalize outside of my sample. Thus, future studies should replicate the surveys with other samples (available on Zenodo) to validate my findings.

The participants in Study IV are not representative across all demographics. The demographics of the respondents are listed in Table 4.3. The majority of participants in survey one are white/European, and many are students. In survey two, participants were primarily of Asian (of Chinese nationality) and white/European descent and many students. When presenting my results, I present proportions based on the total number of respondents in each demographic group. I also used chi-squared tests to determine the significance between different demographic groups, which accounts for the sample size of each population being compared. Therefore, findings found to be statistically significant had a sufficient number of respondents in each demographic to satisfy the test. However, due to a low number of participants in some demographic groups, not all demographics could be analyzed. Future studies should replicate this survey to enable an analysis of additional demographics.

In addition, the majority of participants identified as men or women. I did give participants the option to self-specify gender, but very few participants chose to self-specify. Thus, my analysis was limited to only the differences between participants who identified as men and women. Again, future work should replicate the study to enable analysis beyond these binary genders.

My findings are based only on self-reported feedback habits. Demographic information of feedback givers is not readily available on the feedback channels I investigated (often, the writer's real name is not even given). This data sparsity problem means my findings cannot be directly validated against actual feedback data. One previous study, by Guzman and Rojas (2019), approximated the gender of feedback givers on app stores from their usernames. Using these approximations, they found that men were more likely than women to provide feedback on the Apple app store, which is in line with what the respondents reported and supports the findings.

6. Conclusion

User involvement in IS development is a crucial pillar for building systems that meet users' needs, demands, and desires. With the numerous, heterogeneous, and diverse user groups that use IS in their professional and private lives, involving wide audiences in RE becomes more important than ever. Inspired by the significance of user involvement in IS development, I set out with this thesis to explore how AI-based technologies can support data collection and analysis and understand who creates data (e.g., reviews or comments) in online channels today and for what reasons.

Throughout four primary studies included in this thesis, I present and evaluate two innovative AI-based systems, a RE chatbot and a qualitative coding IML system, and show results from a large-scale study on IS feedback engagement. Study I proposes the first design of a laddering interview chatbot that can conduct exploratory interviews with novice users without requiring prior domain knowledge for its configuration. In Study II, I utilize the chatbot to extract a hierarchical map of user values and positive and negative gains in smartphone use. Therein, I expand and update my understanding of why smartphones are used and compare a chatbot interviewer to established survey-based approaches based on empirical evidence. In Study III, I contribute the design of an IML system developed initially to support the content coding of laddering interview data, but that became a tool for qualitative data in general. Therein, I provide a comprehensive overview of challenges for AI-based approaches to support qualitative coding as well as prescriptive knowledge for designing semi-automated qualitative coding systems. Furthermore, the study proposes concepts for making code rules suggestions, ML-model training with sparse datasets, and explaining AI recommendations to users. The study also contributes descriptive knowledge from the first empirical study of user-generated code rules for qualitative coding. In Study IV, I present results from two survey studies that I conducted with participants from New Zealand, China, and Germany, which shed light on user groups' demographics and motivations that give online feedback. Thereby, I contribute knowledge to understanding the coverage of requirements extracted from online feedback and suggest strategies for encouraging online feedback and more evenly including underrepresented groups.

Overall, this thesis demonstrates the promise of AI-based systems for data collection and analysis from wide audiences. While Ladderbot as an instantiation of a laddering chatbot can be improved in many ways, I could successfully apply the bot for user interviews, with a significant impact on interviewee engagement. Still, the results stemming from survey-based laddering were highly valuable to draw an extensive hierarchical map of smartphone goals and values. For data analysis, AI proved helpful, yet not necessarily for accelerating coding but for coding with higher quality. Nevertheless, improvements in how fast large datasets can be analyzed are required the make wide audience involvement feasible. As such, I am excited to see how Cody and related approaches will be extended in future work. Additionally, with the relevance of explainable AI in various fields extending far beyond qualitative analysis, it will be interesting to see if users desire but ignore explanations in other domains as well. With regards to the coverage of feedback from software users, future studies will show how feedback motivation may not only differ between demographics such as age or gender but between cultures as well. Potentially, these studies may even use chatbot interviewer rather than surveys to engage participants, thus providing researchers with more *field evaluations* of chatbot interviews.

The demand for ISD remains high and undertaking RE with wide audiences becomes increasingly critical. I am hopeful that the designs, architectures, and insights into AI-based data collection and analysis support and into IS feedback behavior provide useful devices for user involvement. I am convinced that with this thesis, I offer a meaningful contribution toward the application of such devices both in research and in practice. Through tools that give underrepresented users and user groups a voice, not only in development projects or organizational matters but societal topics as well, I can create a world that strives to create human-centered software for everyone.

7. Appendix

A Study 1

A.1 Example Chatbot Laddering Interview

The following example shows one of the 85 interviews performed by that laddering chatbot. The transcript was translated to English from the original interview language German. Before the following interview excerpt, the participant provided the chatbot with the three most used functions on their smartphone. Ladderbot questions are printed in cursive characters.

1st attribute: Social Media Apps
*Bot: Your 1st example was *Social Media Apps*. Why do you use *Social Media Apps*? What value does this feature offer you?*
Participant: It connects me with my friends and family
What feeling does this give you?
It gives me a sense of belonging and makes me happy
Why is that?
Because it allows me to live my social life and stay connected to the people I care about.
Do you think media coverage has influenced your usage behavior? In what way?
Yes, since almost everything is digital nowadays, I am almost forced to be active in social networks so that I don't lose touch with the world.
*Is there an emotion that describes your last answer in relation to the *Social Media Apps* feature?*
Affected, Connected
*Imagine that you could not use the *Social Media Apps* feature. What alternatives would you use and why?*
I would have to resort to calling and texting since all my friends/family's smartphones still have this feature. However, that requires a lot more compromise on my part.
Can you think of more to add to your last answer?
Yes, without the social media apps, I would spend much less time using my smartphone.
*Can you think of any specific problems that the *Social Media Apps* feature could cause? Can you think of ways to improve the feature?*
That you spend too much time on these apps and less time in reality. One way would be to limit the usage time daily, for example.
Why is this important to you?
Because with today's technology, you lose a little bit more of your humanity

2nd attribute: Photo and video recording
*Your 2nd example was *Photo and video recording*. Why do you use *Photo and Video Capture*? What is the added value of this feature?*
Because I like to keep physical/electronic memories of moments that are important to me.
What feeling does it give you?

Nostalgia, happiness
Why is this important to you?
Because I like to remember beautiful moments and then share them with my friends/family.
Why is such a thing important for you?
Because then I have the feeling that nothing is forgotten/lost.
*What problems could *photography and videography* lead to? How would *photography and videography* need to change to avoid these problems?*
That you pay less attention to what's in front of you, but I can't think of any "solutions" to that right now
*Has anything ever bothered you about the *Photo and Video Capture* feature or did something not work?*
That maybe the photos didn't look as good as I wanted, or that it took too long to capture the perfect moment
What would be an example of your last answer?
When you see a show, and then in the photos/videos it doesn't look quite the same as in reality
What did you do when you didn't have the *photo and video capture* feature on your smartphone?
Used a separate camera device
How does it help you to be able to do that on your smartphone now?
Having everything on a single device
Would you say that how you use the *Photo and Video Capture* feature has changed in the last few years?
Yes
Do you think your answer sufficiently describes your reasons, or can you think of anything else?
The performance is getting better and you have more options for shots

3rd attribute: Internet Browsing
Your 3rd example was *Internet Browsing*. Why do you use *Internet Browsing*? What added value does this feature give you?
Because I want to stay informed and access information quickly.
Why is this important to you?
To not become ignorant and to be able to understand everything, at least whatever I can.
Do you think that the *Internet Browsing* function could also cause problems? Can you think of solutions to the problems?
That you need less common sense and skills in problem-solving
Why is that?
Because you always need to look up one thing and not think about why something can be like that.
If your smartphone didn't have the *Internet Browsing* feature, what alternatives would you use? Why?
Ask people or search in literature
Could you explain your last answer in more detail?

If you don't have your phone with you but still need the information, you would have to ask friends or go to a library and read books to get the information.
Has this always been your opinion on the subject, or would you have answered this question differently?
It has always been like this
What feeling does this give you?
stop

B Study 2

B.1 Final Codebook

Attributes	**Times Survey/Chatbot**	**Example verbatim**
A101 Mobile commerce	13 (11/2)	*“Online banking“, “Paypal app“*
A102 Management of Schedule and Information	85 (68/17)	*“Calendar”, “Alarm clock”, “Watch”*
A103 Entertainment	179 (113/66)	*“Camera“, “Music streaming“, “YouTube“*
A104 Communication	305 (196/109)	*“WhatsApp“, “Messaging services“, “Calls“*
A105 Information Search	159 (100/59)	*“Browser“, “News services“, “Navigation“*
A106 Social Media	67 (31/36)	*“Social Media“, “Facebook“, “Instagram“*
A107 Basic device features	15 (12/3)	*“Wifi“, “Bluetooth“*
Consequences		
C201 Increased availability flexibility	174 (105/69)	*“Want to respond to important messages quickly and from anywhere”*
C202 Productive personal life	137 (132/105)	*“So that I get all my tasks done by the end of the day”*
C203 Productive work life	41 (31/10)	*“I am often awake or working at night and can thus forward the work results.”*
C204 Simplification of physical tasks and positive substitution	311 (207/104)	*“You don’t need a separate camera to take good photos these days”*
C205 Enable improve communication	526 (292/234)	*“You can always answer when you feel like it and do not have to take the time when it just does not fit”*
C208 Sharing information and data	76 (46/30)	*“I would like to remember it later and let other people share my life this way”*
C209 No negative impact/indifference	244 (0/244)	*“No problems”*
C210 Extend general knowledge and inspiration	270 (162/108)	*“I can also educate myself in my free time on all topics that interest me at the moment”*
C211 Extend social knowledge	75 /30/45)	*“I can understand and follow the activities of other people”*
C212 Digital storage	117 (66/51)	*“I like to capture moments to look at them later”*
C213 Feeling good and being entertained	302 (145/157)	*“Good against boredom and creates happiness and satisfaction”*
C216 Improve health	19 (16/3)	*“Because I want to live healthier and therefore do more sports”*
C217 Source/Risk diversification	6 (1/5)	*“Use different browsers to not provide one provider with all of my data”*
Negative gains		
N301 Spending or wasting (more) time	44 (0/44)	*“You quickly get lost in the app. So you spend too much time on your phone”*
N302 Technology substitution, evasion or downgrade	258 (0/258)	*“I would probably call my families more often and communicate with friends through other platforms like Facebook, etc. via the laptop”*
N303 Misunderstandings/impersonal communication	33 (1/32)	*“Sometimes messages lack the meta-level, which can lead to misunderstandings”*
N304 (Strong) negative feelings	111 (7/104)	*“A sense of addiction and loss of control”*
N305 Feeling unsafe and out of control	41 (0/41)	*“I feel partly observed, because I don’t know who sees my sent photos & co.”*
N306 Inattentiveness/thoughtlessness	57 (0/57)	*“At concerts, I want to enjoy the moment and not be glued to my cell phone”*
N307 Involuntary availability	32 (0/32)	*“Sometimes I turn everything off. It’s annoying anyway when you’re always available”*
N308 Negative health effects	21 (0/21)	*“it makes me a bit of an addict and it has become a bit of a routine”*
N309 Repulsive content and feeling disgusted	23 (0/23)	*“sometimes the uploaded pictures, GIFs and videos are also repulsive and not good for me”*
N310 Unreliable Information and false data	22 (0/22)	*“Too much false information. You should read carefully and decide for yourself what is true and what is not”*
N311 Low service/functionality performance	105 (0/105)	*“I am annoyed by the so-called ”dead spots”, i.e., the lack of network coverage”*
Values		
V401 Convenience	60 (40/20)	*“it is handy to be able to use only one device for several things”*
V402 Self-optimization	110 (107/3)	*“I do not like unnecessary waste of time”*
V403 Socialization	158 (118/40)	*“I don’t want to be alone”*
V404 Unobtrusiveness	18 (2/16)	*“I’m negatively affect other people’s moods, which I don’t want”*
V405 Knowledge	82 (43/39)	*“I like to have a broad general knowledge, even if that is often difficult”*
V406 Hedonism	56 (45/11)	*“I want to have as much fun as possible”*
V407 Sense of comfort	84 (81/3)	*“Reduces stress because you have a reliable tool that stores everything”*
V408 Satisfaction	72 (66/6)	*“I can live my life as planned”*
V409 Safety and privacy	78 (4/74)	*“I can quickly inform other people or get help, especially in emergencies”,“I value my privacy”*
V410 (Mental) health	10 (3/7)	*“one should not become dependent on digital communication”*
V411 Autonomy	30 (23/7)	*“You can stay on schedule and keep a balance between duty and freedom”*
V413 Kinship	42 (9/33)	*“interpersonal relationships play an important role in my psyche”*
Prompts		
Z501 Downsides of a functionality	286 (0/286)	*“I don’t want to rely on Google throughout, I also want to exercise my brain”*
Z502 Function/service unavailable	275 (0/275)	*“I would use the notebook to look for information”*
Z503 Impact of news coverage and development over time	209 (0/209)	*“since almost everything is digital nowadays then I am almost forced to be active in the social networks so that I don’t lose touch with the world”*

B.2 Complete AIM

The following AIM demonstrates complete AIM, including information from all three treatments (survey- and chatbot-based laddering). Active cells (>= cutoff 12) are highlighted in gray.

		<----Consequences---->													<----Negative Gains---->											<----Values---->												<--Prompts-->			In-	Out-			
	#	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	degrees	degrees	Centrality	Abstractness	#
A101 Mobile commerce	0	5	1	1	9			3	3			1			1	2	1									2	2			1	1	1	2			3		2	2	2	0	45	0,006	0,000	0
A102 Management of Schedule and Information	1	9	54	16	43	2	1	24	3		8	7	5		1	12		5	2	2		1			5	6	33	2	2		4	12	13	5	1	3	1	22	16	14	0	334	0,042	0,000	1
A103 Entertainement	2	25	13	3	58	3	24	51	24		65	11	1		12	45		22	8	1		5	5	1	2	7	17	8	6	9	23	29	16	8	5	3		57	6	47	0	800	0,102	0,000	2
A104 Communication	3	7	57	1	59	28	29	81	23	28		31	4	2	1	86	22	41	15	15	22	6	2	1	32	25	15	99	2	5	9	11	16	27	3	12	28	17	11	74	0	1460	0,186	0,000	3
A105 Information Search	4	35	47	4	6	14		47	119	4		19	3	2	5	43	2	17	4	12	4	3	7	14	23	13	22	6	3	47	6	14	15	14	1	7		66	56	46	0	804	0,102	0,000	4
A106 Social Media	5	2	7		8	41	1	26	19	33		35	2	1	7	18	1	13	9	9	2	3	5	2	4	1	3	19	1	5	5	4	4	9		1	11	29	36	24	0	409	0,052	0,000	5
A107 Basic device features	6	5	2	1	8	2		4	4		1	2				2		1	1						3	2	2		1		1	1		5				3	4	2	0	57	0,007	0,000	6
C201 Increased availability & flexibility	7		14	5	13	17	5	5	13	3	6	14						2			1	2				7	13	4		1	1	5	1	6		12					306	177	0,061	0,634	7
C202 Productive personal life	8	8		12	18	19	6	2	6	1	4	1	1				1	1	2		1					4	41	9	2	2	8	18	19	12	1	3	1				382	212	0,076	0,643	8
C203 Productive work life	9	1	2		1								1								1					2	11					4	7	1		1					92	32	0,016	0,742	9
C204 Simplification of physical tasks and positive substitution	10	27	34	8		19	11		2	2	13	19	5			1		1								2	32	14		8	11	2	22	8	1	6	2				444	304	0,095	0,594	10
C205 Enable & improve communication	11	45	54	8	39		38	2	28	37		47	6	1	5	1	6	2	2	2	1	1				16	13	145	1	6	9	14	2	5	1	9	45				533	609	0,145	0,467	11
C208 Sharing information and data	12	2	2	1	8	14			2	5	6	3	1		1											2	3	16		1		3	3				5				159	78	0,030	0,671	12
C209 No negative impact/indifference	13	1	1		2	2						6				4		3			1		1			1	1	1			1			3	1	1					516	30	0,069	0,945	13
C210 Extend general knowledge and inspiration	14	24	29	5	26	24	2	3		12	1	28	5					1								2	15	16	1	65	4	12	2	4		7	2				323	308	0,080	0,512	14
C211 Extend social knowledge	15		2		2	18	2	1	5			11	1		1												3	2		4	2	3	3			1	9				147	88	0,030	0,626	15
C212 Digital storage	16	1	3		8	1	21	2	8			13	1														14	4		1	12	8	1		3						120	110	0,029	0,522	16
C213 Feeling good and being entertained	17	8	12	1	6	13	4	3	22	12	9		1		3		1	5	1							2	22	11	3	9	23	45	11	1	2	4	4				410	238	0,082	0,633	17
C216 Improve health	18			2	2				2			2															6			1	2	1	4		1		2				40	25	0,008	0,615	18
C217 Source/Risk diversification	19				2											1		1	1								1														13	6	0,002	0,684	19
N301 Spending or wasting (more) time	20	1	2			3		1	1							1	1	2	1	2		2						2			2				1						96	22	0,015	0,814	20
N302 Technology substitution, evasion or downgrade	21	8	7	5	1	15		16	3	1	1	5		1	3		2	7	4							1		1		1				7			2				436	100	0,068	0,813	21
N303 Misunderstandings/impersonal communication	22		1			1		3								2		3	3						2			2	2					1			1				63	21	0,011	0,750	22
N304 (Strong) negative feelings	23	1	3	1				2		1		1				5	1		1		2		1		5	3	1				2		2				1				259	33	0,037	0,887	23
N305 Feeling unsafe and out of control	24										1			1	1	1		3		1		1	1	2	5				3	1			1	16		2					103	40	0,018	0,720	24
N306 Inattentiveness/thoughtlesness	25				1	2		1			1	2			4	1	1	4	1			1	1					5	7		1			1			1				105	35	0,018	0,750	25
N307 Involuntary availability	26		1	1		4										1		3	1			2	1									1		2		1	2				61	20	0,010	0,753	26
N308 Negative health effects	27		1			1									3	1		3	1	1	2														2						44	15	0,008	0,746	27
N309 Repulsive content and feeling disgusted	28			2		1												7		1				1	1				2	1											45	16	0,008	0,738	28
N310 Unreliable Information and false data	29																			1										2				2		1					42	6	0,006	0,875	29
N311 Low service/functionality performance	30		1					5	1			1		1	5	2	1	12	3	1			2	2		14								4			1				189	56	0,031	0,771	30
V401 Convenience	31				1	1			2	1		1						2		1					1		1	1				3				4					154	19	0,022	0,890	31
V402 Self-optimization	32		5	2	2	1	2		2				1					2										2			5	8	12			3					288	47	0,043	0,860	32
V403 Socialization	33				3	7	1	2	1	3		3	1													4	3			1	5	6	4		2	1	1				410	57	0,059	0,878	33
V404 Unobtrusiveness	34																																	1		1	1				55	3	0,007	0,948	34
V405 Knowledge	35	1	1		1	1			3			1															2	2	1		1	2	5								190	21	0,027	0,900	35
V406 Hedonism	36				1			1			1	7															2	1				5	1		1						155	20	0,022	0,886	36
V407 Sense of comfort	37	1										2															4		2				1			1					237	11	0,032	0,956	37
V408 Satisfaction	38				2							2	1														4	2	2	1		5									227	19	0,031	0,923	38
V409 Safety and privacy	39	1	2		1	1						2						3	5					1	1	1			2	1	1		2		1	2					186	27	0,027	0,873	39
V410 (Mental) health	40																																1								32	1	0,004	0,970	40
V411 Autonomy	41	1			1													1									1			1	2	1	1								94	9	0,013	0,913	41
V413 Kinship	42	1	1		1	1		1		2		1																1					1								134	19	0,019	0,876	42
Z501 Downsides of a functionality	43	1	2		1	6		63	1		1	5		1	22	12	22	35	32	44	15	16	14	14	81	14	1	5	12	1	3	1	1	26	2	4	2				286	460	0,095	0,383	43
Z502 Function/service unavailable	44	9	11	3	17	13		5	3	1	1	11			2	186		51	2		3				5	5		2		1	1			6	1		2				275	386	0,084	0,416	44
Z503 Impact of news coverage and development over time	45	13	1	1	3	6	3	117	5	1	1	8		3	1	9	1	6	4	3	6	1	5	4	1			1		5	1			12	2	1	1				209	271	0,061	0,435	45

Figure B.1: Complete AIM of treatments PP, VPP, and LB.

B.3 Complete Positive Gains HVM, Cutoff 12

The following figure shows the HVM of positive gains for the cutoff 12. Nodes are ordered by centrality (horizontal order) and abstractness (vertical order). Linkages with high frequency are highlighted.

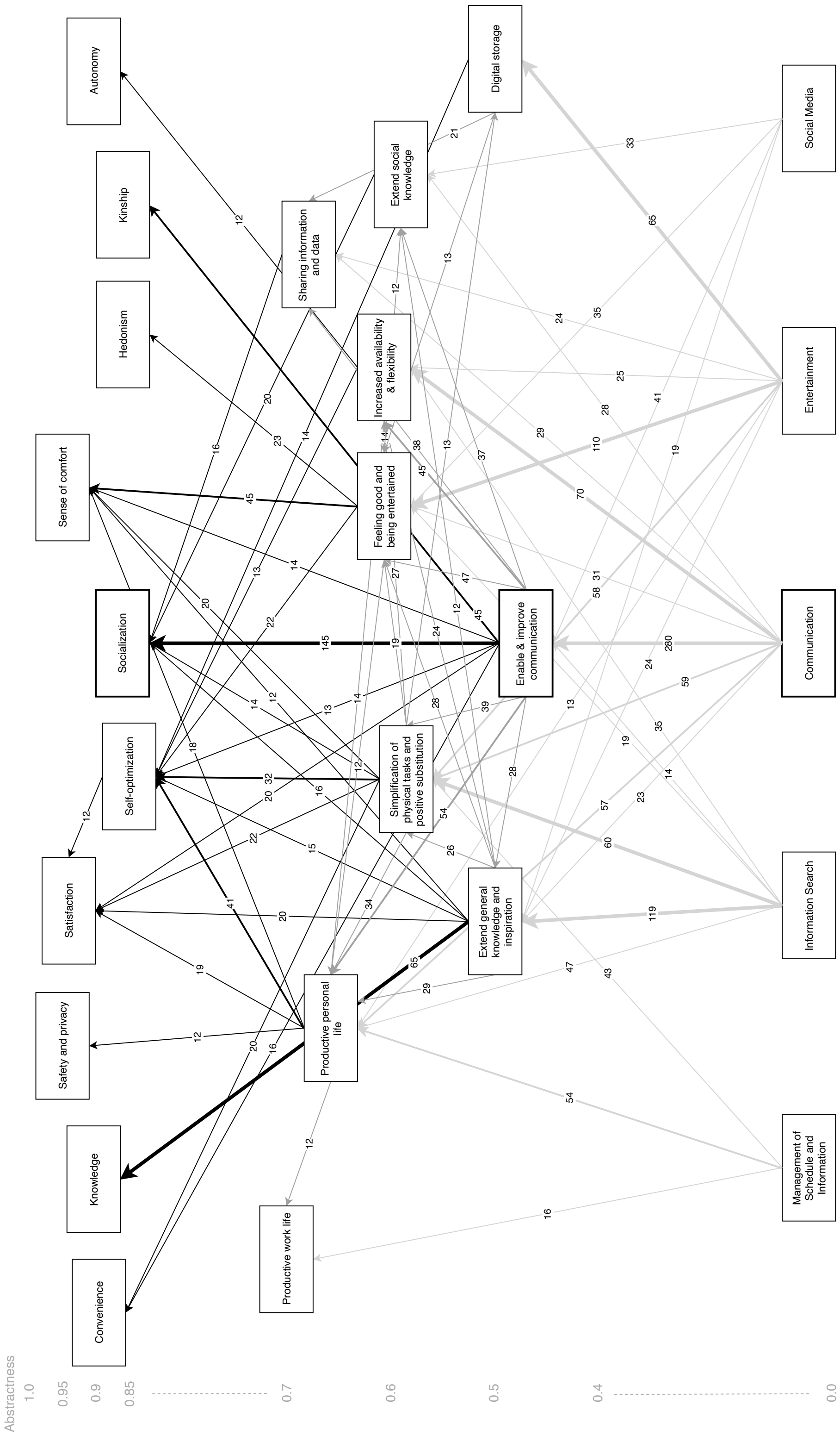


Figure B.2: Complete HVM of positive gains including treatments PP, VPP, and LB.

B.4 Complete Negative Gains HVM, Cutoff 12

The following figure shows the HVM of negative gains for the cutoff 12, with the node downsides of a functionality and some consequences removed for clarity. Nodes are ordered by centrality (horizontal order) and abstractness (vertical order), and linkages with high frequency are highlighted.

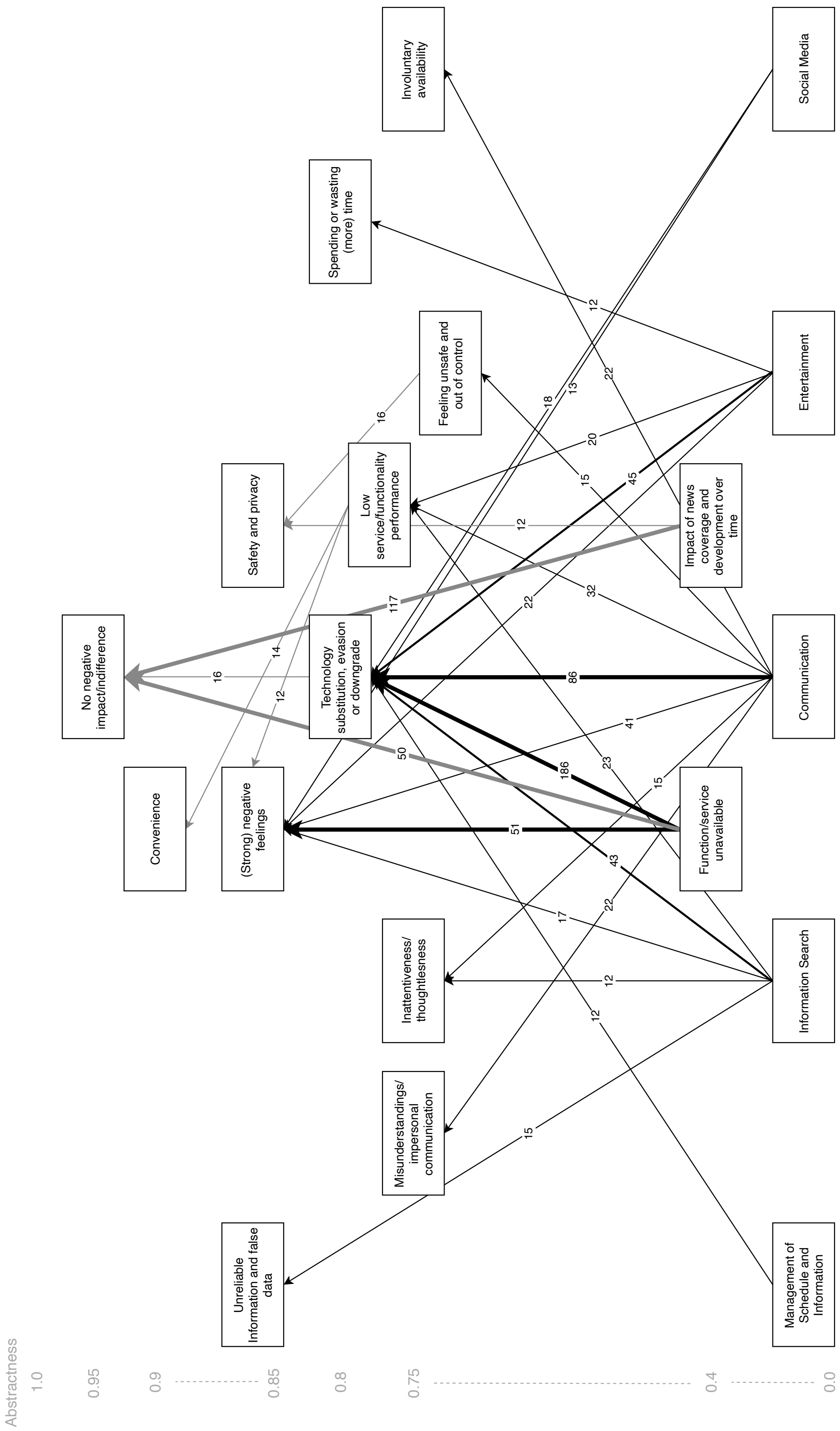


Figure B.3: Complete HVM of negative gains including treatments PP, VPP, and LB.

B.5 Comparison Survey-based and Chatbot-based AIM

The following AIM demonstrates the differences in results from the survey- and chatbot-based laddering. It was created by subtracting linkages from chatbot interviews from linkages from survey interviews. Positive values indicate that the linkage results from survey-based laddering, negative values indicate the origin of the linkage to be chatbot laddering. Active cells are highlighted in gray.

		<---Consequences--->													<---Negative Gains--->											<---Values--->												<--Prompts-->			
	#	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	#
A101 Mobile commerce	0	5	-1	1	7			-3	3			-1			-1	-2	-1									2	2			1	1	1	2			3		-2	-2	-2	0
A102 Management of Schedule and Information	1	1	3	12	25		1	-24	3		6	-1	5		-1	-12		-5	-2	-2		-1			-5	6	33	2	-2		4	12	13	-3	1	1	-1	-22	-16	-14	1
A103 Entertainement	2	-1	1	1	1	-3	6	-51			11	14	1		-12	-45		-18	-8	-1		-5	-5	-1	-2	-1	17	6	-4	5	17	27	12	-8	-3	1		-57	-6	-47	2
A104 Communication	3	1	7	2	15	66	9	-81	17	-6		-17	2		-1	-86	-2	-37	-15	-15	-22	-6	-2	-1	-32	11	15	47	-2	1	7	9	14	-23	-1	6	-14	-17	-11	-74	3
A105 Information Search	4	15	-1	2	1	2		-47	27	2		-11	3	-2	-5	-43	-2	-17	-4	-12	-4	-3	-7	-14	-23	-1	18	4	-1	-7	4	14	13	-14	-1	5		-66	-56	-46	4
A106 Social Media	5	-2	-7		-4	-11	-2	-26	-3	-3		-5	-2	-1	-7	-18	-1	-11	-9	-9	-2	-3	-5	-2	-4	-1	3	3	-1	1	-3	2	2	-9		-1	-9	-29	-36	-24	5
A107 Basic device features	6	5		1	6	2		-4	4		1	2				-2		-1	-1						-3		2		-1		1	1		-3				-3	-4	-2	6
C201 Increased availability & flexibility	7		2	3	9	5	3	-5	3	3	2	-2						-2			-1	-2				7	11	2		4	6	5	1	-2		8					7
C202 Productive personal life	8			12	2	-7	2	-2	4	-1	2	-8	1				1	-1	-2		-1					4	41	7	-2		6	18	19	-1	-1	3	-1				8
C203 Productive work life	9	1	2		-1								1								-1					2	11					4	7	1		1					9
C204 Simplification of physical tasks and positive substitution	10	11	4	6		9	7		8		13	-3	5			-1		1								2	32	8		4	11	2	22	-6	1	6					10
C205 Enable & improve communication	11	17	6	2	25		4	-2	14	-9		-29		1	-5	-1	-4		-2	-2	-1	-1				16	13	61	-1	-2	9	14	18	-3	1	7	-29				11
C208 Sharing information and data	12	2		1	2	-2			2	1	-4	-1	-1		-1											2	3	4		1		3	3				-5				12
C209 No negative impact/indifference	13	-1	-1		-2	-2						-6				-4		-3			-1		-1			-1	-1	-1			-1			-3	-1	-1					13
C210 Extend general knowledge and inspiration	14	12	11	5	16	2	-2	-3		-4	-1	-14	-1					1								2	11	8	1	-9	4	12	16	-2		3	-2				14
C211 Extend social knowledge	15		2			-8	-2	-1	-3			-7	-1		-1												3	-2		-2	2	3	3			1	-9				15
C212 Digital storage	16	1	1		4	-1	-1	-2	-2			-5	1														14	4		1	8	4	1		-3						16
C213 Feeling good and being entertained	17	2		1	-2	-3		-3	-4	6	-1		1		-3		-1	5	-1							2	22	1	1	5	13	41	3	-1		4	-4				17
C216 Improve health	18			2	2																						6			1	2	1	4		1		-2				18
C217 Source/Risk diversification	19				-2											-1		1	-1								1														19
N301 Spending or wasting (more) time	20	-1	-2			-3		-1	-1							-1	-1	-2	-1	-2		-2						-2			-2				-1						20
N302 Technology substitution, evasion or downgrade	21	-8	-7	-5	-1	-15		-16	-3	-1	-1	-5		-1	-3		-2	-7	-4							-1		-1		-1				-7			-2				21
N303 Misunderstandings/impersonal communication	22		-1			-1		-3								-2		-3	-3						-2			-2	-2					-1			-1				22
N304 (Strong) negative feelings	23	-1	-3	-1				-2		-1		-1				-5	-1		-1		-2		-1		-5	-3	1				-2		2				-1				23
N305 Feeling unsafe and out of control	24										-1			-1	-1	-1		-3		-1		-1	-1	-2	-5				-3	-1			-1	-16		-2					24
N306 Inattentiveness/thoughtlesness	25				-1	-2		-1			-1	-2			-4	-1	-1	-4	-1			-1	-1					-5	-7		-1			-1			-1				25
N307 Involuntary availability	26		-1	-1		-4										-1		-3	-1			-2	-1									-1		-2		-1	-2				26
N308 Negative health effects	27		-1			-1									-3	-1		-3	-1	-1	-2														-2						27
N309 Repulsive content and feeling disgusted	28			-2		-1												-7		-1				-1	-1				-2	-1											28
N310 Unreliable Information and false data	29																			-1										-2				-2		-1					29
N311 Low service/functionality performance	30		-1					-5	-1			-1		-1	-5	-2	-1	-12	-3	-1			-2	-2		-14								-4			-1				30
V401 Convenience	31				1	1				1		-1								-1					-1		1	1				3				4					31
V402 Self-optimization	32		5	2	2	1	2		2				1					2										2			5	8	12			3					32
V403 Socialization	33				3	5	1	-2	1	-1		1	1													4	3			1	5	6	4			1	8				33
V404 Unobtrusiveness	34																																	-1		1	-1				34
V405 Knowledge	35	1	1		1	-1			-1			1																	1		1	2	3								35
V406 Hedonism	36				1			-1			-1	3															2	1				5	1		-1						36
V407 Sense of comfort	37	1										2															4		2				1			1					37
V408 Satisfaction	38				2								1														4	-2	2	1		5									38
V409 Safety and privacy	39	-1	-2		-1	-1						-2						-3	-5					-1	-1	1			-2	-1	-1		-2		-1	-2					39
V410 (Mental) health	40																																1								40
V411 Autonomy	41	1			1													-1									1			-1	2	1	1								41
V413 Kinship	42	-1	-1		-1	-1		-1		-2		-1																-8					-1								42
Z501 Downsides of a functionality	43	-1	-2		-1	-6		-63	-1		-1	-5		-1	-22	-12	-22	-35	-32	-44	-15	-16	-14	-14	-81	-14	-1	-5	-12	-1	-3	-1	-1	-26	-2	-4	-2				43
Z502 Function/service unavailable	44	-9	-11	-3	-17	-13		-5	-3	-1	-1	-11			-2	-186		-51	-2		-3				-5	-5		-2		-1	-1			-6	-1		-2				44
Z503 Impact of news coverage and development over time	45	-13	-1	-1	-3	-6	-3	-117	-5	-1	-1	-8		-3	-1	-9	-1	-6	-4	-3	-6	-1	-5	-4	-1			-1		-5	-1			-12	-2	-1	-1				45

Figure B.4: Comparison of survey-based and chatbot-based AIMs for smartphone values.

C Study 3

C.1 Formative Study: Interview Guide

I used the following interview guide during the formative evaluation of the Cody prototype. Constructs used in the survey are inspired and adapted from W. Wang and Benbasat (2005). I used a tool to translate the interview guide from German to English for this presentation.

Interview Guide
Cody: Formative Evaluation Study

First Stage: Problem Awareness and Perception of Tool
Sample: Researcher with training in qual. research

I. General information

- Discipline of research
- What kinds of studies are conducted (Research or Evaluation)?
- How often are qualitative studies performed
- Which methodology is used (grounded theory, inductive, iterative…)?
  - What kind of research questions need to be solved
  - In what form is data collected and analyzed
- What tools are used to perform collection / analysis

II. Perception of Cody tool

Features

- Most liked feature (Perceived Ease of Use)
- Most disliked feature (Perceived Ease of Use)
- Which features are missing to support coding? (Perceived Usefulness)
  - How do you feel about unit-of-analysis functionality?
  - Is any feature unnecessary?

Support functions

- Work with coding rules? How is that integrated usually?
  - Do you feel comfortable with revising the code rules? (Perceived Ease of Use)
- Can you communicate your coding needs to the system (Trust - Competence)

- Perception of interface? Easy, complex? (Perceived Ease of Use)
- Do you think it would help to speed up you coding? (Perceived usefulness)
- Do you think it would help to identify errors? (Perceived usefulness)
  - How integrated would you want to be into ML model training?
  - Cody has the ability to consider your needs and preferences (Trust - Competence)

Trust / Explainability

- Are the suggestions transparent, is Cody honest? (Trust – Integrity)
  - How are the suggestions made?
- Is the prediction of the behavior of the system possible? (Trust – Integrity)
  - What information would you need in addition to understand the suggestions?
- Cody keeps your interest in mind (Trust – Benevolence)
- Cody puts your interests first; process does feel unintentional / unnatural? (Trust – Benevolence)

III. Comparison to regular coding

- Was the process "easier" than the coding you already did
  - Are there differences in your approach to coding with Cody?
- What are concerns you have regarding using Cody?
  - How might Cody change how you code your transcripts?
  - How might Cody have affected the speed of your coding (Perceived usefulness)
  - How might Cody have affected the quality of your codes (Perceived usefulness)
- Could you share your coded data for this particular section with me, so I can compare it against your coding of this section?

IV. Qualitative Research method and process

Coding Process

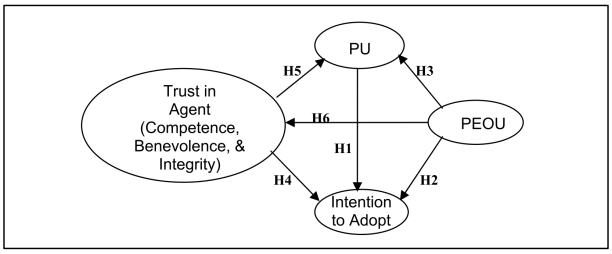
- What is a typical coding process?
  - How is coding started
  - How are codes organized? Why is organization important
- Units-of-analysis considerations
  - How are multiple coders integrated?

- What is the most interesting in coding?
  - What is the most tedious in coding?
  - How do you use data from first pass coding for further analysis

Automation

- Willingness to use software that (partially) automated coding (Intention to Adopt)
  - Would you use Cody in your research? (Intention to Adopt)
- How should the support look like?
- Condition for such assistance to be appreciated
  - When in the coding process would such assistance be appreciated
  - Previous experiences with coding support / automation

Constructs adapted from the following research model for trust in and adoption of online recommendation agents.



W. Wang and I. Benbasat, "Trust in and Adoption of Online Recommendation Agents," *J. Assoc. Inf. Syst.*, vol. 6, no. 3, pp. 72–101, 2005

Figure C.1: Interview guide: Formative study.

C.2 Summative Study: Interview Guides

I used the following interview guide during the summative evaluation of Cody. This guide contains questions for the Cody treatment. I used a tool to translate the interview guide from German to English for this presentation.

Interview Guide
Cody: Summative Evaluation Study
Treatment Cody (CY)

I. **Basic Information**
  - Course of studies
  - Examples of previous work in the area of qual. coding? What was it about?
  - Frequency of qualitative studies
  - Which methodology (grounded theory, inductive, iterative...)?
    - Was a similar approach used as in this study?
  - Which tools are used?
  - Comparison to 'regular' coding
  - How long did it take you to code? Does that time include only coding or also preparation?
  - How did you like the experiment?
  - How did you like the coding process?
  - What did you find most interesting, what did you find most complex?
  - Was the process different from what you are used to?
    - Did the use of the tool affect you in any way?
    - Which changes in your behavior did Cody require?
    - Was the perceived effort of the process different?
  - Perception of Cody

II. **Introduction**
  - Do you have the feeling to have understood all functions of Cody?
  - Introduction comprehensive enough?
  - Did you take your time to experiment with the functions?
    - Code rules?
    - ML?

III. **Functions**
  - Most liked feature *(Perceived Ease of Use)*
  - Least liked feature *(Perceived Ease of Use)*
  - Which features are missing? *(Perceived Usefulness)*
    - Perception coding support?
    - Any feature unnecessary?

IV. **Recommender**
  - Code rules, New or common? Did it help?
    - Perception Revision of the rules *(Perceived Ease of Use)*
  - Can all needs be transmitted to the system *(Trust - Competence)*
  - Perception of the interface? *(Perceived Ease of Use)*
  - Does the tool help you code faster? *(Perceived Usefulness)*

- Can the tool help to increase the quality of the codes? *(Perceived Usefulness)*
    - How much do you want to perceive the ML support?
    - The tool is able to realize your wishes and needs? *(Trust - Competence)*

V. Trust / Explanations

- Is the tool transparent and sincere? *(Trust - Integrity)*
    - Can you explain why certain suggestions were made?
- Can you predict how the tool will react? *(Trust - Integrity)*
    - What additional information would you need to understand the tool?
- Cody serves your interests *(Trust - Benevolence)*
- Cody adapts to your way of working, or does the process feel unnatural? *(Trust - Benevolence)*
- Do you have reservations about Cody?
    - How could Cody change your coding speed *(Perceived Usefulness)*
    - How could Cody change your coding quality *(Perceived Usefulness)*
    - How could Cody change your behavior?

VI. Automation

- Would you use tools that (partially) automate your coding? *(Intention to Adopt)*
    - Would you use Cody? *(Intention to Adopt)*
- What would tools look like to support you?
    - When during coding would such tools be interesting?
    - What experience do you have with such tools?

Figure C.2: Interview guide: Summative study. Treatment Cody.

I used the following interview guide during the summative evaluation of Cody. This guide contains questions for the MAXQDA treatment. I used a tool to translate the interview guide from German to English for this presentation.

Interview Guide
Cody: Summative Evaluation Study
Treatment MAXQDA (MX)

I. **Basic Information**
   - Course of studies
   - Examples of previous work in the area of qual. coding? What was it about?
   - Frequency of qualitative studies
   - Which methodology (grounded theory, inductive, iterative...)?
     - Was a similar approach used as in this study?
   - Which tools are used?

   - Comparison to 'regular' coding
   - How long did it take you to code? Does that time include only coding or also preparation?
   - How did you like the experiment?
   - How did you like the coding process?
   - What did you find most interesting, what did you find most complex?
   - Was the process different from what you are used to?
     - Did the use of the tool affect you in any way?
     - Which changes in your behavior did MAXQDA require?
     - Was the perceived effort of the process different?

   - Perception of MAXQDA

II. **Introduction**

   - Do you have the feeling to have understood all functions of MAXQDA?
   - Introduction comprehensive enough?
   - Did you take your time to experiment with the functions?
     - Did you look at the tutorial?

III. **Functions**

   - Most liked feature *(Perceived Ease of Use)*
   - Least liked feature *(Perceived Ease of Use)*
   - Which features are missing? *(Perceived Usefulness)*
     - Perception coding support?
     - Any feature unnecessary?

IV. **Recommender**

   - Can all needs be transmitted to the system *(Trust - Competence)*

   - Perception of the interface*? (Perceived Ease of Use)*
   - Does the tool help you code faster? *(Perceived Usefulness)*

V. Trust / Explanations

- Is the tool transparent and sincere? *(Trust - Integrity)*
- Can you predict how the tool will react? *(Trust - Integrity)*
  - What additional information would you need to understand the tool?
- MAXQDA adapts to your way of working, or does the process feel unnatural? *(Trust - Benevolence)*

VI. Automation

- Would you use tools that (partially) automate your coding? *(Intention to Adopt)*
  - Would you use MAXQDA? *(Intention to Adopt)*
- What would tools look like to support you?
  - When during coding would such tools be interesting?
  - What experience do you have with such tools?
- Did you see the introduction of the other treatment? Are you interested in more support while coding, interested in using the alternative tool?

Figure C.3: Interview guide: Summative study. Treatment MAXQDA.

Bibliography

Abbasi, A. (2016). Big Data Research in Information Systems: Toward an Inclusive Research Agenda. *Journal of the Association for Information Systems*, *17*(2), 1–32. https://doi.org/10.1017/CBO9781107415324.004

Abdul-Kader, S. A., & Woods, J. (2015). Survey on Chatbot Design Techniques in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications*, *6*(7). https://doi.org/10.14569/IJACSA.2015.060712

Abrami, G., Lücking, A., Mehler, A., Rieb, E., & Helfrich, P. (2019). TEXTANNOTATOR : A flexible framework for semantic annotations. *Proceedings of the 15th Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-15)*, 1–12.

Ahmad, M. U., Hanna, A., Mohamed, A.-Z., Schlindwein, A., Pley, C., Bahner, I., Mhaskar, R., Pettigrew, G. J., & Jarmi, T. (2019). A Systematic Review of Opt-out Versus Opt-in Consent on Deceased Organ Donation and Transplantation (2006–2016). *World Journal of Surgery*, 1–11. https://doi.org/10.1007/s00268-019-05118-4

Appan, R., & Browne, G. (2012). The Impact of Analyst-Induced Misinformation on the Requirements Elicitation Process. *MIS Quarterly*, *36*(1), 85–106.

Bagozzi, R. P., & Dabholkar, P. A. (1994). Consumer recycling goals and their effect on decisions to recycle: A means-end chain analysis. *Psychology & Marketing*, *11*(4). https://doi.org/10.1002/mar.4220110403

Bakharia, A., Bruza, P., Watters, J., Narayan, B., & Sitbon, L. (2016). Interactive Topic Modeling for aiding qualitative content analysis. *CHIIR 2016 - Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval*, 213–222. https://doi.org/10.1145/2854946.2854960

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*. https://doi.org/10.1037/0033-295X.84.2.191

Bano, M., Zowghi, D., & da Rimini, F. (2018). User Involvement in Software Development: The Good, the Bad, and the Ugly. *IEEE Software*, *35*(6), 8–11. https://doi.org/10.1109/MS.2018.4321252

Bano, M., Zowghi, D., Ferrari, A., Spoletini, P., & Donati, B. (2018). Learning from mistakes: An empirical study of elicitation interviews performed by novices. *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, 182–193. https://doi.org/10.1109/RE.2018.00027

Basit, T. N. (2003). Manual or electronic? The role of coding in qualitative data analysis. *Educational Research*, *45*(2), 143–154. https://doi.org/10.1080/0013188032000133548

Beck, K., & Fowler, M. (2000). *Planning Extreme Programming*.

Bednar, P. M., & Welch, C. (2009). Contextual Inquiry and Requirements Shaping. *Information systems development* (pp. 225–236). Springer US. https://doi.org/10.1007/978-0-387-68772-8{_}18

Bleize, D. N., & Antheunis, M. L. (2019). Factors influencing purchase intent in virtual worlds: A review of the literature. *Journal of Marketing Communications*, *25*(4), 403–420.

Bock, O., Nicklisch, A., & Baetge, I. (2012). Hroot: Hamburg registration and organization online tool. *WiSo-HH Working Paper Series*.

Bødker, M., Gimpel, G., & Hedman, J. (2014). Time-out/time-in: The dynamics of everyday experiential computing devices. *Information Systems Journal*, *24*(2), 143–166. https://doi.org/10.1111/isj.12002

Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., & Madachy, R. (1998). Using the WinWin spiral model: A case study. *Computing Practices*, *31*(7), 33–44. https://doi.org/10.1109/2.689675

Botschen, G., Thelen, E. M., & Pieters, R. (2004). Using means-end structures for benefit segmentation. *European Journal of Marketing*, *33*(1/2), 38–58. https://doi.org/10.1108/eum0000000004491

Bourne, H., & Jenkins, M. (2005). Eliciting managers' personal values: An adaptation of the laddering interview method. *Organizational Research Methods*, *8*(4), 410–428. https://doi.org/10.1177/1094428105280118

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77–101.

Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, *61*, 163–181. https://doi.org/10.1016/j.infsof.2015.01.004

Browne, G. J., & Rogich, M. B. (2001). An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. *Journal of Management Information Systems*, *17*(4), 223–249. https://doi.org/10.1080/07421222.2001.11045665

Buhrmester, M. D., Kwang, T., & Gosling, S. (2011). Amazon's mechanical turk. *Perspectives on Psychological Science*, *6*, 3–5.

Burnett, M., Stumpf, S., Macbeth, J., Makri, S., Beckwith, L., Kwan, I., Peters, A., & Jernigan, W. (2016). Gendermag: A method for evaluating software's gender inclusiveness. *Interacting with Computers*, *28*(6), 760–787.

Chakraborty, S., Sarker, S., & Sarker, S. (2010). An exploration into the process of requirements elicitation: A grounded approach. https://doi.org/10.17705/1jais.00225

Chaput, M. (2020). Whoosh. https://whoosh.readthedocs.io/en/latest/index.html

Chen, C., Zhang, K. Z., Gong, X., Lee, M. K., & Wang, Y. (2020). Decreasing the problematic use of an information system: An empirical investigation of smartphone game players. *Information Systems Journal*, *30*(3), 492–534. https://doi.org/10.1111/isj.12264

Chen, C.-H., Khoo, L. P., & Yan, W. (2002). A strategy for acquiring customer requirement patterns using laddering technique and ART2 neural network. *Advanced Engineering Informatics*, *16*(3), 229–240. https://doi.org/10.1016/S1474-0346(03)00003-X

Chen, C.-H., Trappey, A. C., Peruzzini, M., Stjepandić, J., & Wognum, N. (2016). *Advances in Transdisciplinary Engineering* (Vol. 4).

Chen, N.-C., Brooks, M., Kocielnik, R., Hong, S. (, Smith, J., Lin, S., Qu, Z., & Aragon, C. (2017). Lariat: A Visual Analytics Tool for Social Media Researchers to Explore Twitter Datasets. *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*, 1881–1890. https://doi.org/10.24251/hicss.2017.228

Chen, N.-C., Drouhard, M., Kocielnik, R., Suh, J., & Aragon, C. R. (2018). Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. *ACM Transactions on Interactive Intelligent Systems*, *8*(2), 1–21. https://doi.org/10.1145/3185515

Chen, N.-C., Kocielnik, R., Drouhard, M., Suh, J., Cen, K., Zheng, X., Aragon, C. R., & Pena-Araya, V. (2016). Challenges of Applying Machine Learning to Qualitative Coding. *ACM SIGCHI Workshop on Human-Centered Machine Learning.*

Chen, N., Lin, J., Hoi, S. C. H., Xiao, X., & Zhang, B. (2014). Ar-miner: Mining informative reviews for developers from mobile app marketplace. *Proceedings of the 36th International Conference on Software Engineering*, 767–778. https://doi.org/10.1145/2568225.2568263

Cheng, H. F., Wang, R., Zhang, Z., O'Connell, F., Gray, T., Harper, F. M., & Zhu, H. (2019). Explaining decision-making algorithms through UI: Strategies to help non-expert stakeholders. *Proceedings of the Conference on Human Factors in Computing Systems (CHI '19)*, 1–12. https://doi.org/10.1145/3290605.3300789

Chiu, C. M. (2005). Applying means-end chain theory to eliciting system requirements and understanding users perceptual orientations. *Information and Management*, *42*(3), 455–468. https://doi.org/10.1016/j.im.2004.02.002

Chiu, C. M., Wang, E. T., Fang, Y. H., & Huang, H. Y. (2014). Understanding customers' repeat purchase intentions in B2C e-commerce: The roles of utilitarian value, hedonic value and perceived risk. *Information Systems Journal*, *24*(1), 85–114. https://doi.org/10.1111/j.1365-2575.2012.00407.x

Chun, H., Lee, H., & Kim, D. (2012). The integrated model of smartphone adoption: Hedonic and utilitarian value perceptions of smartphones among Korean college students. *Cyberpsychology, Behavior, and Social Networking*, *15*(9), 473–479. https://doi.org/10.1089/cyber.2012.0140

Cohn, M. (2004). *User Stories Applied: For Agile Software Development.* Pearson Education, Inc. www.wowebook.com

Collins, E., Rozanov, N., & Zhang, B. (2019). *LIDA: Lightweight Interactive Dialogue Annotator*. https://doi.org/10.18653/v1/d19-3021

Coulin, C. R. (2007). *A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation* (Doctoral dissertation). University of Technology, Sydney.

Crossler, R. E., & Bélanger, F. (2019). Why would I use location-protective settings on my smartphone? Motivating protective behaviors and the existence of the privacy knowledge–belief gap. *Information Systems Research*, *30*(3), 995–1006. https://doi.org/10.1287/isre.2019.0846

Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, *15*(6), 523–543. https://doi.org/10.1080/13645579.2011.625764

Crowston, K., Liu, X., & Allen, E. E. (2010). Machine Learning and Rule-Based Automated Coding of Qualitative Data. *Proceedings of the American Society for Information Science and Technology*, *47*(1), 1–2. https://doi.org/https://doi.org/10.1002/meet.14504701328

De Almeida, C. A., Freitas, F., Costa, A. P., & Moreira, A. (2019). WEBQDA: The Quest for a Place in the Competitive World of CAQDAS. *Proceedings of the 2019 International Conference on Engineering Applications (ICEA)*. https://doi.org/10.1109/CEAP.2019.8883456

De Oliveira, G. F., Ferreira, B., & Marques, A. B. (2020). USARP method: Eliciting and describing USAbility Requirements with Personas and user stories. *ACM International Conference Proceeding Series*, 437–446. https://doi.org/10.1145/3422392.3422435

Debnath, S., & Spoletini, P. (2020). Designing a Virtual Client for Requirements Elicitation Interviews. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12045 LNCS*, 160–166. https://doi.org/10.1007/978-3-030-44429-7{_}12

Derrick, D. C., Read, A., Nguyen, C., Callens, A., & De Vreede, G. J. (2013). Automated group facilitation for gathering wide audience end-user requirements. *Annual Hawaii International Conference on System Sciences (HICSS'13)*, 195–204. https://doi.org/10.1109/HICSS.2013.109

Dery, K., Kolb, D., & Maccormick, J. (2014). Working with connective flow: How smartphone use is evolving in practice. *European Journal of Information Systems*, *23*(5), 558–570. https://doi.org/10.1057/ejis.2014.13

Deutsch, S., Begolli, G., Lugmayr, M., & Tscheligi, M. (2011). Assisted collection and organization for laddering interview data. *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, 647. https://doi.org/10.1145/1979742.1979661

Dieste, O., & Juristo, N. (2011). Systematic review and aggregation of empirical studies on elicitation techniques. *IEEE Transactions on Software Engineering*, *37*(2), 283–304. https://doi.org/10.1109/TSE.2010.33

Dieste, O., Lopez, M., & Ramos, F. (2008). Updating a systematic review about selection of software requirements elicitation techniques. *11th Workshop on Requirements Engineering, WER 2008 - Proceedings*, 96–103.

Ding, F., & Liu, B. (2011). Effects of user participation in enterprise system improvement on service value perceived by users. *PACIS 2011 - 15th Pacific Asia Conference on Information Systems: Quality Research in Pacific*, 1–14.

Drouhard, M., Chen, N. C., Suh, J., Kocielnik, R., Pena-Araya, V., Cen, K., Zheng, X., & Aragon, C. R. (2017). Aeonium: Visual analytics to support collaborative qualitative coding. *IEEE Pacific Visualization Symposium*, 220–229. https://doi.org/10.1109/PACIFICVIS.2017.8031598

Duarte, D., Farinha, C., da Silva, M. M., & da Silva, A. R. (2012). Collaborative Requirements Elicitation with Visualization Techniques. *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 343–348. https://doi.org/10.1109/WETICE.2012.14

Dudley, J. J., & Kristensson, P. O. (2018). A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems*, *8*(2). https://doi.org/10.1145/3185517

Dumitru, A. I., Girbacia, T., Boboc, R. G., Postelnicu, C. C., & Mogan, G. L. (2018). Effects of smartphone based advanced driver assistance system on distracted driving behavior: A simulator study. *Computers in Human Behavior*. https://doi.org/10.1016/j.chb.2018.01.011

Eickhoff, M., & Wieneke, R. (2018). Understanding Topic Models in Context: A Mixed-Methods Approach to the Meaningful Analysis of Large Document Collections. *Proceedings of the 51st Hawaii International Conference on System Sciences*, 903–912. https://doi.org/10.24251/hicss.2018.113

Elrakaiby, Y., Ferrari, A., Spoletini, P., Gnesi, S., & Nuseibeh, B. (2017). Using Argumentation to Explain Ambiguity in Requirements Elicitation Interviews. *Proceedings - 2017 IEEE 25th International Requirements Engineering Conference, RE 2017*, 51–60. https://doi.org/10.1109/RE.2017.27

Etikan, I. (2016). Comparison of convenience sampling and purposive sampling. *American Journal of Theoretical and Applied Statistics*, *5*, 1. https://doi.org/10.11648/j.ajtas.20160501.11

Evers, J. C. (2018). Current issues in qualitative data analysis software (QDAS): A user and developer perspective. *Qualitative Report*, *23*(13), 61–73.

Fan, M., Li, Y., & Truong, K. N. (2020). Automatic Detection of Usability Problem Encounters in Think-aloud Sessions. *ACM Transactions on Interactive Intelligent Systems*, *10*(2), 1–24. https://doi.org/10.1145/3385732

Feine, J., Morana, S., & Maedche, A. (2019). Designing a Chatbot Social Cue Configuration System. *Proceedings of the International Conference on Information Systems*.

Ferreira, B., Silva, W., Barbosa, S. D., & Conte, T. (2018). Technique for representing requirements using personas: A controlled experiment. *IET Software*, *12*(3). https://doi.org/10.1049/iet-sen.2017.0313

Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *Interactions*, *24*(4), 38–42. https://doi.org/10.1145/3085558

Freitas, F., Ribeiro, J., Brandão, C., de Souza, F. N., Costa, A. P., & Reis, L. P. (2018). In case of doubt see the manual: A comparative analysis of (self)learning packages qualitative research software. *Advances in Intelligent Systems and Computing*, *621*, 176–192. https://doi.org/10.1007/978-3-319-61121-1{_}16

Friesen, E., Bäumer, F. S., & Geierhos, M. (2018). CORDULA: Software requirements extraction utilizing chatbot as communication interface. *CEUR Workshop Proceedings*, *2075*(Section 2), 1–5.

Galitz, W. O. (2007). *The essential guide to user interface design: An introduction to gui design principles and techniques*. John Wiley & Sons.

Ganji, A., Orand, M., & McDonald, D. W. (2018). Ease on down the code: Complex collaborative qualitative coding simplified with 'Code wizard'. *Proceedings of the ACM on Human-Computer Interaction*. https://doi.org/10.1145/3274401

Gao, S., Li, Y., & Guo, H. (2019). Understanding the Value of Using Smartphones for Older Adults in China: A Value-Focused Thinking Approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11701 LNCS*, 533–544. https://doi.org/10.1007/978-3-030-29374-1{_}43

García-López, D., Segura-Morales, M., & Loza-Aguirre, E. (2020). Improving the quality and quantity of functional and non-functional requirements obtained during requirements elicitation stage for the development of e-commerce mobile applications: An alternative reference process model. *IET Software*, *14*(2), 148–158. https://doi.org/10.1049/iet-sen.2018.5443

Garrity, E. J. (2001). Synthesizing User Centered and Designer Centered IS Development Approaches Using General Systems Theory. *Information Systems Frontiers*, *3*(1), 107–121. https://doi.org/10.1023/A:1011457822609

Gasson, S. (2003). Human-Centered vs. User-Centered Approaches to Information System Design. *Journal of Information Technology Theory and Application*, *5*(2), 29–46.

Glinz, M. (2005). Rethinking the Notion of Non-Functional Requirements. *Proceedings of the Third World Congress for Software Quality*, (September), 55–64.

Glinz, M. (2007). On Non-Functional Requirements. *15th IEEE International Requirements Engineering Conference (RE 2007)*, 21–26. https://doi.org/10.1109/RE.2007.45

Gnewuch, U., Morana, S., Adam, M. T., & Maedche, A. (2018). Faster is not always better: Understanding the effect of dynamic response delays in human-chatbot interaction. *26th European Conference on Information Systems: Beyond Digitization - Facets of Socio-Technical Change (ECIS 2018).*

Gnewuch, U., Morana, S., & Maedche, A. (2017). Towards Designing Cooperative and Social Conversational Agents for Customer Service. *Proceedings of the 2017 International Conference on Information Systems (ICIS'17)*, 1–13.

Goel, V., & Pirolli, P. (1989). Motivating the notion of generic design within information-processing theory: the design problem space. *AI Magazine*, *10*(1).

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, *21*(3), 267–297. https://doi.org/10.1093/pan/mps028

Groen, E. C., Seyff, N., Ali, R., Dalpiaz, F., Doerr, J., Guzman, E., Hosseini, M., Marco, J., Oriol, M., Perini, A., et al. (2017). The crowd in requirements engineering: The landscape and challenges. *IEEE software*, *34*(2), 44–52.

Grünbacher, P., & Boehm, B. (2001). EasyWinWin: A Groupware-supported methodology for requirements negotiation. *Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering - ESEC/FSE-9*, 320. https://doi.org/10.1145/503209.503265

Günther, W. A., Rezazade Mehrizi, M. H., Huysman, M., & Feldberg, F. (2017). Debating big data: A literature review on realizing value from big data. *Journal of Strategic Information Systems*, *26*(3), 191–209. https://doi.org/10.1016/j.jsis.2017.07.003

Guo, Y., & Barnes, S. (2007). Why people buy virtual items in virtual worlds with real money. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, *38*(4), 69–76.

Gutman, J. (1982). A Means-End Chain Model Based on Consumer Categorization Processes. *Journal of Marketing*. https://doi.org/10.2307/3203341

Guzman, E., Alkadhi, R., & Seyff, N. (2016). A needle in a haystack: What do twitter users say about software? *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 96–105. https://doi.org/10.1109/RE.2016.67

Guzman, E., Alkadhi, R., & Seyff, N. (2017). An exploratory study of twitter messages about software applications. *Requirements Engineering*. https://doi.org/10.1007/s00766-017-0274-x

Guzman, E., Ibrahim, M., & Glinz, M. (2017). A little bird told me: Mining tweets for requirements and software evolution. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 11–20.

Guzman, E., Oliveira, L., Steiner, Y., Wagner, L. C., & Glinz, M. (2018). User feedback in the app store: A cross-cultural study. *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, 13–22.

Guzman, E., & Rojas, A. P. (2019). Gender and user feedback: An exploratory study. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 381–385.

Harding, J. (2015). *Identifying Themes and Coding Interview Data: Reflective Practice in Higher Education*. SAGE Publications, Ltd. https://doi.org/10.4135/9781473942189

Hassan, N. R., & Mathiassen, L. (2017). Distilling a body of knowledge for information systems development. *Information Systems Journal*, *28*(1), 175–226. https://doi.org/10.1111/isj.12126

Hedman, J., Bødker, M., Gimpel, G., & Damsgaard, J. (2019). Translating evolving technology use into user stories: Technology life narratives of consumer technology use. *Information Systems Journal*, *29*(6), 1178–1200. https://doi.org/10.1111/isj.12232

Heim, G. R., Wentworth, W. R., & Peng, X. (2009). The value to the customer of RFID in service applications. *Decision Sciences*, *40*(3), 477–512. https://doi.org/10.1111/j.1540-5915.2009.00237.x

Heinze, J., Thomann, M., & Fischer, P. (2017). Ladders to m-commerce resistance: A qualitative means-end approach. *Computers in Human Behavior*, *73*, 362–374. https://doi.org/10.1016/j.chb.2017.03.059

Henfridsson, O., & Lindgren, R. (2010). User involvement in developing mobile and temporarily interconnected systems. *Information Systems Journal*, *20*(2), 119–135. https://doi.org/10.1111/j.1365-2575.2009.00337.x

Hickey, A. M., & Davis, A. M. (2004). A unified model of requirements elicitation. *Journal of Management Information Systems*, *20*(4), 65–84. https://doi.org/10.1080/07421222.2004.11045786

Ho, C. Y., Li, P. C., Young, S. T., & Lai, Y. H. (2020). Efficacy of a Smartphone Hearing Aid Simulator. *Journal of Medical and Biological Engineering*, *40*(4), 496–504. https://doi.org/10.1007/s40846-020-00519-6

Hofmann, H. F., & Lehner, F. (2001). Requirements Engineering as a Success Factor in Software Projects. *IEEE Software*, *18*(4), 58–66.

Huang, I.-L., & Burns, J. R. (2000). A Cognitive Comparison of Modelling Behaviors Between Novice and Expert Information Analysts. *Sixth Americas Conference on Information Systems (AMCIS 2000)*, 1316–1322.

Huang, T. H., Chang, J. C., & Bigham, J. P. (2018). Evorus: A Crowd-powered conversational assistant built to automate itself over time. *UIST 2017 Adjunct - Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, *2018-April*, 155–157. https://doi.org/10.1145/3173574.3173869

Hunter, M. G. (1997). The use of RepGrids to gather interview data about information systems analysts. *Information Systems Journal*, *7*(1), 67–81. https://doi.org/10.1046/j.1365-2575.1997.00005.x

ISCED, U. (2012). International standard classification of education 2011.

Jarzebowicz, A., & Polocka, K. (2017). Selecting requirements documentation techniques for software projects: A survey study. *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017*, 1189–1198. https://doi.org/10.15439/2017F387

Jean-Charles, N., & Spoletini, P. (2019). Developing A Comprehensive Tool to Support Requirements Analyst During Elicitation Interviews. *Journal of Student Research*. https://doi.org/10.47611/jsr.vi.714

Jeon, J., Croft, W. B., Lee, J. H., & Park, S. (2006). A framework to predict the quality of answers with non-textual features. *Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 228–235.

Jia, J., & Capretz, L. F. (2018). Direct and mediating influences of user-developer perception gaps in requirements understanding on user participation. *Requirements Engineering*, *23*(2), 277–290. https://doi.org/10.1007/s00766-017-0266-x

Jipeng, Q., Zhenyu, Q., Yun, L., Yunhao, Y., & Xindong, W. (2019). Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. *Journal of Latex Class Files*, *14*(8), 1–17. http://arxiv.org/abs/1904.07695

Johnson, D., Tizard, J., Damian, D., Blincoe, K., & Clear, T. (2020). Open crowdre challenges in software ecosystems. *2020 4th International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*, 1–4. https://doi.org/10.1109/CrowdRE51214.2020.00007

Jolly, J. P., Reynolds, T. J., & Slocum, J. W. (1988). Application of the means-end theoretic for understanding the cognitive bases of performance appraisal. *Organizational Behavior and Human Decision Processes*, *41*(2), 153–179. https://doi.org/10.1016/0749-5978(88)90024-6

Jung, Y. (2014). What a smartphone is to me: Understanding user values in using smartphones. *Information Systems Journal*, *24*(4), 299–321. https://doi.org/10.1111/isj.12031

Jung, Y., & Kang, H. (2010). User goals in social virtual worlds: A means-end chain approach. *Computers in Human Behavior*, *26*(2), 218–225. https://doi.org/10.1016/j.chb.2009.10.002

Kaciak, E., & Cullen, C. W. (2009). A method of abbreviating a laddering survey. *Journal of Targeting, Measurement and Analysis for Marketing*, *17*(2), 105–113. https://doi.org/10.1057/jt.2009.4

Kalpokaite, N., & Radivojevic, I. (2018). Best practice article: Auto-coding and Smart Coding in ATLAS.ti Cloud. https://atlasti.com/2018/09/27/auto-coding-and-smart-coding-in-atlas-ti-cloud/

Kassel, N. W., & Malloy, B. A. (2003). An Approach to Automate Requirements Elicitation and Specification. *Proc. of the 7th Int. Conf. on Software Engineering and Applications*, 544–549.

Kato, J., Komiya, S., Saeki, M., Ohnishi, A., Nagata, M., Yamamoto, S., & Horai, H. (2001). A model for navigating interview processes in requirements elicitation. *Proceedings of the Asia-Pacific Software Engineering Conference and International Computer Science Conference, APSEC and ICSC*, 141–148. https://doi.org/10.1109/APSEC.2001.991470

Kehr, F., Kowatsch, T., Wentzel, D., & Fleisch, E. (2015). Blissfully ignorant: The effects of general privacy concerns, general institutional trust, and affect in the privacy calculus. *Information Systems Journal*, *25*(6), 607–635. https://doi.org/10.1111/isj.12062

Keith, M. J., Babb, J. S., Lowry, P. B., Furner, C. P., & Abdullat, A. (2015). The role of mobile-computing self-efficacy in consumer information disclosure. *Information Systems Journal*, *25*(6), 637–667. https://doi.org/10.1111/isj.12082

Kelly, D., Wacholder, N., Rittman, R., Sun, Y., Kantor, P., Small, S., & Strzalkowski, T. (2007). Using interview data to identify evaluation criteria for interactive, analytical question-answering systems. *Journal of the American Society for Information Science and Technology*, *58*(7), 1032–1043. https://doi.org/10.1002/asi.20575

Khan, J. A., Xie, Y., Liu, L., & Wen, L. (2019). Analysis of requirements-related arguments in user forums. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 63–74.

Kim, D., Chun, H., & Lee, H. (2014). Determining the factors that influence college students' adoption of smartphones. *Journal of the Association for Information Science and Technology*, *65*(3), 578–588. https://doi.org/10.1002/asi.22987

Kim, J. H., Ham, S. M., Cha, H. J., & Kim, K. K. (2018). Selection of requirement elicitation techniques using laddering. *2017 4th International Conference on Systems and Informatics, ICSAI 2017*, *2018-Janua*(Icsai), 1604–1609. https://doi.org/10.1109/ICSAI.2017.8248540

Kim, K. M., & Hwang, J. H. (2020). Factors affecting smartphone online activity use in South Korea: with a focus on the moderating effect of disability status. *Universal Access in the Information Society*, (0123456789). https://doi.org/10.1007/s10209-020-00758-z

Kim, S., Chang, J. J. E., Park, H. H., Song, S. U., Cha, C. B., Kim, J. W., & Kang, N. (2020). Autonomous Taxi Service Design and User Experience. *International Journal of Human-Computer Interaction*, *36*(5), 429–448. https://doi.org/10.1080/10447318.2019.1653556

Kim, S., Lee, J., & Gweon, G. (2019). Comparing Data from Chatbot and Web Surveys. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 1–12. https://doi.org/10.1145/3290605.3300316

Klein, H. (2003). Crisis in the IS Field? A Critical Reflection on the State of the Discipline. *Journal of the Association for Information Systems*, *4*(1), 237–294. https://doi.org/10.17705/1jais.00037

Klie, J.-C., Bugert, M., Boullosa, B., de Castilho, R. E., & Gurevych, I. (2018). The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. *Proceedings of the International Conference on Computational Linguistics*, 5–9.

Klopfenstein, L. C., Delpriori, S., Malatini, S., & Bogliolo, A. (2017). The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms. *2017 Conference on Designing Interactive Systems (DIS'17)*, 555–565. https://doi.org/10.1145/3064663.3064672

Knäble, M., Nadj, M., & Maedche, A. (2019). Oracle or Teacher ? A Systematic Overview of Research on Interactive Labeling for Machine Learning. *Internationaler Kongress Für Wirtschaftsinformatik 2020.*

Kolcaba, K. Y., & Kolcaba, R. J. (1991). An analysis of the concept of comfort. *Journal of Advanced Nursing*. https://doi.org/10.1111/j.1365-2648.1991.tb01558.x

Krippendorff, K. (2004). *Reliability in Content Analysis: Some Common Misconceptions and Recommendations* (tech. rep.). http://repository.upenn.edu/ascpapers/242

Kuem, J., Ray, S., Hsu, P. F., & Khansa, L. (2020). Smartphone Addiction and Conflict: An Incentive-Sensitisation Perspective of Addiction for Information Systems. *European Journal of Information Systems*, *00*(00), 1–22. https://doi.org/10.1080/0960085X.2020.1803154

Kuisma, T., Laukkanen, T., & Hiltunen, M. (2007). Mapping the reasons for resistance to Internet banking: A means-end approach. *International Journal of Information Management*, *27*(2), 75–85. https://doi.org/10.1016/j.ijinfomgt.2006.08.006

Kujala, S., & Väänänen-Vainio-Mattila, K. (2009). Value of Information Systems and Products: Understanding the Users' Perspective and Values. *Journal of Information Technology Theory and Application*, *9*(4), 23–39.

Kung, F. Y., Kwok, N., & Brown, D. J. (2018). Are Attention Check Questions a Threat to Scale Validity? *Applied Psychology*, *67*(2), 264–283. https://doi.org/10.1111/apps.12108

Kwon, H. E., So, H., Han, S. P., & Oh, W. (2016). Excessive dependence on mobile social apps: A rational addiction perspective. *Information Systems Research*, *27*(4), 919–939. https://doi.org/10.1287/isre.2016.0658

Ladd, D. A., Datta, A., Sarker, S., & Yu, Y. (2010). Trends in mobile computing within the IS discipline: A ten-year retrospective. *Communications of the Association for Information Systems*. https://doi.org/10.17705/1cais.02717

Lappas, T., Sabnis, G., & Valkanas, G. (2016). The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry. *Information Systems Research*, *27*(4). https://doi.org/10.1287/isre.2016.0674

Leimeister, J. M., Österle, H., & Alter, S. (2014). Digital services for consumers. *Electronic Markets*, *24*(4), 255–258. https://doi.org/10.1007/s12525-014-0174-6

Lejeune, C. (2011). An Illustration of the Benefits of Cassandre for Qualitative Analysis. *Forum: Qualitative Sozialforschung / Forum: Qualitative Social Research [FQS]*, *12*(1), 19. https://doi.org/10.17169/12.1.1513

Levina, & Vaast. (2005). *The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems* (Vol. 29). https://doi.org/10.2307/25148682

Lewis, S. C., Zamith, R., & Hermida, A. (2013). Content Analysis in an Era of Big Data. *Journal of Broadcasting & Electronic Media*, *57*(1), 34–52. https://doi.org/10.1080/08838151.2012.76170

Li, K., Dewar, R. G., & Pooley, R. J. (2005). Computer-Assisted and Customer-Oriented Requirements Elicitation. *2005 13th IEEE International Conference on Requirements Engineering (RE'05)*, 4–5.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Lim, S., Henriksson, A., & Zdravkovic, J. (2021). Data-Driven Requirements Elicitation: A Systematic Literature Review. *SN Computer Science*, *2*(1). https://doi.org/10.1007/s42979-020-00416-4

Lin, C. Y., Chao, Y. C., & Tang, T. W. (2017). Why not be "smarter"? Examining the factors that influence the behavioral intentions of non-smartphone users. *Industrial Management and Data Systems*, *117*(1), 32–49. https://doi.org/10.1108/IMDS-07-2015-0319

Lin, C. F., Fu, C. S., & Chi, T. H. (2020). Constructing a hybrid hierarchical value map to understand young people's perceptions of social networking sites. *Behaviour and Information Technology*, *39*(2), 150–166. https://doi.org/10.1080/0144929X.2019.1589576

Lin, H. W., & Lin, Y. L. (2014). Digital educational game value hierarchy from a learners' perspective. *Computers in Human Behavior*, *30*(1), 1–12. https://doi.org/10.1016/j.chb.2013.07.034

Lin, Y. L., & Lin, H. W. (2011). A study on the goal value for massively multiplayer online role-playing games players. *Computers in Human Behavior*, *27*(6), 2153–2160. https://doi.org/10.1016/j.chb.2011.06.009

Lin, Y. L., Lin, H. W., & Hung, T. T. (2015). Value hierarchy for Massive Open Online Courses. *Computers in Human Behavior*, *53*, 408–418. https://doi.org/10.1016/j.chb.2015.07.006

Lindberg, A. (2020). Developing theory through integrating human and machine pattern recognition. *Journal of the Association for Information Systems*, *21*(1), 90–116. https://doi.org/10.17705/1jais.00593

Lucassen, G., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2016). Improving agile requirements: the Quality User Story framework and tool. *Requirements Engineering*, *21*(3), 383–403. https://doi.org/10.1007/s00766-016-0250-x

Maalej, W., & Nabil, H. (2015). Bug report, feature request, or simply praise? on automatically classifying app reviews. *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, *00*, 116–125. https://doi.org/10.1109/RE.2015.7320414

Maalej, W., Nayebi, M., Johann, T., & Ruhe, G. (2016). Toward Data-Driven Requirements Engineering. *IEEE Software*, *33*(1), 48–54. https://doi.org/10.1109/MS.2015.153

Maedche, A., Botzenhardt, A., & Neer, L. (2013). *Software for People: Fundamentals, Trends and Best Practices*. https://doi.org/10.1007/978-3-642-31371-4

Maedche, A., Legner, C., Benlian, A., Berger, B., Gimpel, H., Hess, T., Hinz, O., Morana, S., & Söllner, M. (2019). AI-Based Digital Assistants: Opportunities, Threats, and Research Perspectives. *Business and Information Systems Engineering*, *61*(4), 535–544. https://doi.org/10.1007/s12599-019-00600-8

Mao, J. Y., Vredenburg, K., Smith, P. W., & Carey, T. (2005). The state of user-centered design practice. https://doi.org/10.1145/1047671.1047677

Marathe, M., & Toyama, K. (2018). Semi-automated coding for Qualitative research: A user-centered inquiry and initial prototypes. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'18)*, 1–12. https://doi.org/10.1145/3173574.3173922

Marder, B., Gattig, D., Collins, E., Pitt, L., Kietzmann, J., & Erz, A. (2019). The avatar's new clothes: Understanding why players purchase non-functional items in free-to-play games. *Computers in Human Behavior*, *91*, 72–83.

Martens, D., & Maalej, W. (2019). Towards understanding and detecting fake reviews in app stores. *Empirical Software Engineering*, *24*(6), 3316–3355.

MAXQDA. (2020). Keyword-in-Context | MAXQDA. https://www.maxqda.de/hilfe-mx20-dictio/keyword-in-context

McCracken, N., Yan, J. S. Y., & Crowston, K. (2014). Design of an Active Learning System with Human Correction for Content Analysis. *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, 59–62. https://doi.org/10.3115/v1/W14-3109

McDonald, N., Schoenebeck, S., & Forte, A. (2019). Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. *Proceedings of the ACM on Human-Computer Interaction*, *3*(CSCW). https://doi.org/10.1145/3359174

McHugh, M. L. (2013). The chi-square test of independence. *Biochemia medica: Biochemia medica*, *23*(2), 143–149.

McTear, M. F. (2002). Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys*, *34*(1), 1–80. https://doi.org/10.1145/505282.505285

Meade, A. W., & Craig, S. B. (2012). Identifying careless responses in survey data. *Psychological Methods*, *17*(3), 437–455. https://doi.org/10.1037/a0028085

Meth, H., Brhel, M., & Maedche, A. (2013). The state of the art in automated requirements elicitation. *Information and Software Technology*, *55*(10), 1695–1709. https://doi.org/10.1016/j.infsof.2013.03.008

Meth, H., Mueller, B., & Maedche, A. (2015). Designing a requirement mining system. *Journal of the Association for Information Systems*, *16*(9), 799–837. https://doi.org/10.17705/1jais.00408

Meza Martínez, M. A., Nadj, M., & Maedche, A. (2019). Towards an Integrative Theoretical Framework of Interactive Machine Learning Systems. *Proceedings of the 27th European Conference on Information Systems (ECIS2019)*, (2014), 1–19.

Miles, S., & Rowe, G. (2004). The Laddering Technique. In G. M. Breakwell (Ed.), *Doing social psychology research* (1st ed., pp. 305–343). The British Psychological Society; Blackwell Publishing Ltd. https://doi.org/10.1002/9780470776278.ch1

Mohedas, I., Daly, S. R., & Sienko, K. H. (2015). Requirements Development: Approaches and Behaviors of Novice Designers. *Journal of Mechanical Design*, *137*(7), 1–10. https://doi.org/10.1115/1.4030058

Moitra, A., Siu, K., Crapo, A., Chamarthi, H., Durling, M., Li, M., Yu, H., Manolios, P., & Meiners, M. (2018). Towards development of complete and conflict-free requirements. *2018 IEEE 26th International Requirements Engineering Conference (RE'18)*, 286–296. https://doi.org/10.1109/RE.2018.00036

Morrison, J., & George, J. F. (1995). Exploring the Software Engineering Component in MIS Research. *Communications of the ACM*, *38*(7), 80–91. https://doi.org/10.1145/213859.214802

Moshagen, M., & Thielsch, M. T. (2010). Facets of visual aesthetics. *International Journal of Human Computer Studies*, *68*(10), 689–709. https://doi.org/10.1016/j.ijhcs.2010.05.006

Mulvey, M. S., Olson, J. C., Celsi, R. L., & Walker, B. A. (1994). Exploring the Relationships between Means End Knowledge and Involvement. *Advances in Consumer Research*, *21*, 51–57.

Muresan, A., & Pohl, H. (2019). Chats with Bots: Balancing Imitation and Engagement. *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI'19 Extended Abstracts)*, 1–6. https://doi.org/10.1145/3290607.3313084

Myers, M. D., & Newman, M. (1999). The qualitative interview in IS research: Examining the craft. *Communication of the Association for Information Systems*, *2*(December).

Nah, F. F.-H., Siau, K., & Sheng, H. (2005). The value of mobile applications. *Communications of the ACM*, *48*(2), 85–90. https://doi.org/10.1145/1042091.1042095

Nakatoh, T., Uchida, S., Ishita, E., & Oga, T. (2016). Automated generation of coding rules: Text-mining approach to ISO 26000. *Proceedings - 2016 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016*, 154–158. https://doi.org/10.1109/IIAI-AAI.2016.210

Nardi, B. a., Whittaker, S., & Bradner, E. (2000). Interaction and outeraction: instant messaging in action. *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00*, 79–88. https://doi.org/10.1145/358916.358975

Neetu Kumari, S., & Pillai, A. S. (2013). A survey on global requirements elicitation issues and proposed research framework. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 554–557. https://doi.org/10.1109/ICSESS.2013.6615370

Nemcova, A., Jordanova, I., Varecka, M., Smisek, R., Marsanova, L., Smital, L., & Vitek, M. (2020). Monitoring of heart rate, blood oxygen saturation, and blood pressure using a smartphone. *Biomedical Signal Processing and Control*, *59*, 101928. https://doi.org/10.1016/j.bspc.2020.101928

New zealand census [Accessed: December 2019]. (2018). https://www.stats.govt.nz/information-releases/2018-census-population-and-dwelling-counts

Newman, J. C., Des Jarlais, D. C., Turner, C. F., Gribble, J., Cooley, P., & Paone, D. (2002). The differential effects of face-to-face and computer interview modes. *American Journal of Public Health*. https://doi.org/10.2105/AJPH.92.2.294

Nunamaker, J. F., Derrick, D. C., Elkins, A. C., Burgoon, J. K., & Patton, M. W. (2011). Embodied Conversational Agent-Based Kiosk for Automated Interviewing. *Journal of Management Information Systems*, *28*(1), 17–48. https://doi.org/10.2753/MIS0742-1222280102

Nvivo. (2020). NVivo 11 - Automatic coding in document sources. http://help-nv11.qsrinternational.com/desktop/procedures/automatic_coding_in_document_sources.html

OpenSignal. (2016). Global State of Mobile Networks. https://www.opensignal.com/reports/2016/08/global-state-of-the-mobile-network

Oriol, M., Stade, M., Fotrousi, F., Nadal, S., Varga, J., Seyff, N., Abello, A., Franch, X., Marco, J., & Schmidt, O. (2018). FAME: Supporting continuous requirements elicitation by combining user feedback and monitoring. *2018 IEEE 26th International Requirements Engineering Conference (RE'18)*, 217–227. https://doi.org/10.1109/RE.2018.00030

Pagano, D., & Maalej, W. (2013). User feedback in the appstore: An empirical study. *2013 21st IEEE International Requirements Engineering Conference (RE)*, 125–134. https://doi.org/10.1109/RE.2013.6636712

Pai, P., & Arnott, D. C. (2013). User adoption of social networking sites: Eliciting uses and gratifications through a means–end approach. *Computers in Human Behavior*, *29*(3), 1039–1053. https://doi.org/10.1016/j.chb.2012.06.025

Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G., & Gall, H. C. (2016). Ardoc: App reviews development oriented classifier. *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 1023–1027. https://doi.org/10.1145/2950290.2983938

Papadopoulos, N., Martın, O. M., Cleveland, M., & Laroche, M. (2011). Identity, demographics, and consumer behaviors. *International Marketing Review*.

Paredes, P., Ferreira, A. R., Schillaci, C., Yoo, G., Karashchuk, P., Xing, D., Cheshire, C., & Canny, J. (2017). Inquire: Large-scale early insight discovery for qualitative research. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 1562–1575. https://doi.org/10.1145/2998181.2998363

Park, J., & Han, S. H. (2013). Defining user value: A case study of a smartphone. *International Journal of Industrial Ergonomics*, *43*(4), 274–282. https://doi.org/10.1016/j.ergon.2013.04.005

Park, J., & Han, S. H. (2018). A value sampling method for evaluating user value: A case study of a smartphone. *International Journal of Mobile Communications*, *16*(4), 440–458. https://doi.org/10.1504/IJMC.2018.092667

Park, Y., & Chen, J. V. (2007). Acceptance and adoption of the innovative use of smartphone. *Industrial Management and Data Systems*, *107*(9), 1349–1365. https://doi.org/10.1108/02635570710834009

Patton, M. Q. (2002). *Qualitative Research & Evaluation Methods: Integrating theory and practice.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python.* https://scikit-learn.org/stable/index.html

Pedrero-Pérez, E. J., Morales-Alonso, S., Rodríguez-Rives, E., Díaz-Olalla, J. M., Álvarez-Crespo, B., & Benítez-Robredo, M. T. (2019). Smartphone nonusers: Associated sociodemographic and health variables. *Cyberpsychology, Behavior, and Social Networking*, *22*(9), 597–603. https://doi.org/10.1089/cyber.2019.0130

Peffers, K., & Gengler, C. E. (2003). How to identify new high-payoff information systems for the organization. *Communications of the ACM*, *46*(1), 83–88. https://doi.org/10.1145/602421.602424

Peffers, K., Gengler, C. E., & Tuunanen, T. (2003a). Extending Critical Success Factors Methodology to Facilitate Broadly Participative Information Systems Planning. *Journal of Management Information Systems*, *20*(1), 51–85. https://doi.org/10.1080/07421222.2003.11045757

Peffers, K., Gengler, C. E., & Tuunanen, T. (2003b). Methodology to Facilitate Broadly Participative Information Systems Planning. *Journal of Management Information Systems*, *20*(1), 51–85.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pérez, F., & Valderas, P. (2009). Allowing end-users to actively participate within the elicitation of pervasive system requirements through immediate visualization. *2009 4th International Workshop on Requirements Engineering Visualization (REV)*, 31–40. https://doi.org/10.1109/REV.2009.1

Pickard, M. D., & Roster, C. A. (2020). Using computer automated systems to conduct personal interviews: Does the mere presence of a human face inhibit disclosure?

Computers in Human Behavior, *105*(May 2019), 106197. https://doi.org/10.1016/j.chb.2019.106197

Pickard, M. D., Schuetzler, R. M., Valacich, J., & Wood, D. A. (2017). Next-Generation Accounting Interviewing: A Comparison of Human and Embodied Conversational Agents (ECAs) as Interviewers. *SSRN Electronic Journal*, (April), 1–21. https://doi.org/10.2139/ssrn.2959693

Pieters, R., Baumgartner, H., & Allen, D. (1995). A means-end chain approach to consumer goal structures. *International Journal of Research in Marketing*. https://doi.org/10.1016/0167-8116(95)00023-U

Pieters, R., Bottschen, G., & Thelen, E. (1998). Customer desire expectations about service employees: An analysis of hierarchical relations. *Psychology and Marketing*, *15*(8), 755–773.

Pompedda, F., Antfolk, J., Zappalà, A., & Santtila, P. (2017). A combination of outcome and process feedback enhances performance in simulations of child sexual abuse interviews using avatars. *Frontiers in Psychology*, *8*(SEP), 1474. https://doi.org/10.3389/fpsyg.2017.01474

Qualtrics [Accessed: December 2019]. (2019). https://www.qualtrics.com

Rajender Kumar Surana, C. S., Shriya, Gupta, D. B., & Shankar, S. P. (2019). Intelligent Chatbot for Requirements Elicitation and Classification. *Proceedings of the 4th IEEE International Conference on Recent Trends on Electronics, Information, Communication and Technology (RTEICT 2019)*, 866–870. https://doi.org/10.1109/RTEICT46194.2019.9016907

Rashid, A., Meder, D., Wiesenberger, J., & Behm, A. (2006). Visual requirement specification in end-user participation. *First International Workshop on Multimedia Requirements Engineering, MeRE'06*. https://doi.org/10.1109/MERE.2006.7

Reddivari, S., Chen, Z., & Niu, N. (2012). ReCVisu: A tool for clustering-based visual exploration of requirements. *2012 20th IEEE International Requirements Engineering Conference, RE 2012 - Proceedings*. https://doi.org/10.1109/RE.2012.6345828

Reynolds, T. J., Dethloff, C., & Westberg, S. (2001). Advancements in Laddering. *Understanding consumer decision making* (p. 27).

Reynolds, T. J., & Gutman, J. (1988). Laddering theory, method, analysis, and interpretation. *Journal of Advertising Research*, *28*(1), 11–31.

Richards, L. (2002). Qualitative computing—a methods revolution? *International Journal of Social Research Methodology*, *5*(3), 263–276. https://doi.org/10.1080/13645570210146302

Richardson, M., Hussain, Z., & Griffiths, M. D. (2018). Problematic smartphone use, nature connectedness, and anxiety. *Journal of Behavioral Addictions*, *7*(1), 109–116. https://doi.org/10.1556/2006.7.2018.10

Rietz, T. (2019). Designing a Conversational Requirements Elicitation System for End-Users. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 452–457. https://doi.org/10.1109/RE.2019.00061

Rietz, T., & Maedche, A. (2019). LadderBot: A Requirements Self-Elicitation System. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 357–362. https://doi.org/10.1109/RE.2019.00045

Rietz, T., & Maedche, A. (2020). Towards the Design of an Interactive Machine Learning System for Qualitative Coding. *Proceedings of the 41st International Conference on Information Systems (ICIS 2020)*, 0–9. https://doi.org/10.5445/IR/1000124563

Rietz, T., & Maedche, A. (2021a). Cody: An AI-Based System to Semi-Automate Coding for Qualitative Research. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2021)*. https://doi.org/10.1145/3411764.3445591

Rietz, T., & Maedche, A. (2021b). Re-Evaluating User Values of Smartphones - A Wide Audience Qualitative Research Study Comparing Survey- and Chatbot-Based Laddering Interviews. *Under review at: Information Systems Journal (ISJ)*.

Rietz, T., & Schneider, F. (2020). We see we disagree: Insights from Designing a Cooperative Requirements Prioritization System. *European Conference on Information Systems (ECIS)*, (May).

Rietz, T., Toreini, P., & Maedche, A. (2020). Cody: An Interactive Machine Learning System for Qualitative Coding. *UIST 2020 - Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 90–92. https://doi.org/10.1145/3379350.3416195

Rinderle, J., & Hoover, S. P. (1990). Function and form relationships : strategies for preliminary design. http://repository.cmu.edu/meche

Ritchie, J., & Lewis, J. (2003). *Qualitative Research Practice: A guide for social science students and researchers* (Vol. 1). https://doi.org/10.18352/jsi.39

Rugg, G., Eva, M., Mahmood, A., Rehman, N., Andrews, S., & Davies, S. (2002a). Culture Via Laddering. *Information Systems Journal*, *12*(4), 215–229. https://doi.org/10.1046/j.1365-2575.2002.00124.x

Rugg, G., Eva, M., Mahmood, A., Rehman, N., Andrews, S., & Davies, S. (2002b). Eliciting information about organizational culture via laddering. *Information Systems Journal*, *12*(4), 215–229. https://doi.org/10.1046/j.1365-2575.2002.00124.x

Russell, C. G., Busson, A., Flight, I., Bryan, J., van Lawick van Pabst, J. A., & Cox, D. N. (2004). A comparison of three laddering techniques applied to an example of a complex food choice. *Food Quality and Preference*, *15*(6), 569–583. https://doi.org/10.1016/j.foodqual.2003.11.007

Russell, C. G., Flight, I., Leppard, P., van Lawick van Pabst, J. A., Syrette, J. A., & Cox, D. N. (2004). A comparison of paper-and-pencil and computerised methods of ”hard” laddering. *Food Quality and Preference*, *15*(3), 279–291. https://doi.org/10.1016/S0950-3293(03)00068-5

Rzepka, C. (2019). Examining the use of voice assistants: A value-focused thinking approach. *25th Americas Conference on Information Systems, AMCIS 2019*, (September).

Sánchez-Gómez, M. C., Martín-Cilleros, M. V., & Sánchez Sánchez, G. (2019). Evaluation of Computer Assisted Qualitative Data Analysis Software (CAQDAS) Applied to Research. *Springer* (pp. 474–485). https://doi.org/10.1007/978-3-030-20798-4{_}41

Sarker, S., Xiao, X., & Beaulieu, T. (2013). Qualitative studies in information systems: A critical review and some guiding principles. *MIS Quarterly: Management Information Systems*, *37*(4), 3–18.

Schrunner, S., Geiger, B. C., Zernig, A., & Kern, R. (2020). A generative semi-supervised classifier for datasets with unknown classes. *Proceedings of the ACM Symposium on Applied Computing*, 1066–1074. https://doi.org/10.1145/3341105.3373890

Schultze, U., & Avital, M. (2011). Designing interviews to generate rich data for information systems research. *Information and Organization*, *21*(1), 1–16. https://doi.org/10.1016/j.infoandorg.2010.11.001

Sheng, H., Nah, F. F. H., & Siau, K. (2005). Strategic implications of mobile technology: A case study using Value-Focused Thinking. *Journal of Strategic Information Systems*, *14*(3), 269–290. https://doi.org/10.1016/j.jsis.2005.07.004

Shin, D. H., & Choo, H. (2012). Exploring cross-cultural value structures with smartphones. *Journal of Global Information Management*, *20*(2), 67–93. https://doi.org/10.4018/jgim.2012040104

Snijders, R., Dalpiaz, F., Brinkkemper, S., Hosseini, M., Ali, R., & Özüm, A. (2015). REfine: A gamified platform for participatory requirements engineering. *1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE'15)*, 1–6. https://doi.org/10.1109/CrowdRE.2015.7367581

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, *34*(1), 233–272. https://doi.org/10.1023/A:1007562322031

Sorbo, A. D., Panichella, S., Alexandru, C. V., Visaggio, C. A., & Canfora, G. (2017). Surf: Summarizer of user reviews feedback. *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 55–58. https://doi.org/10.1109/ICSE-C.2017.5

Spears, J. L., & Barki, H. (2010). User Participation in Information Systems Security Risk Management. *MIS Quarterly*, *34*(3), 503–522. https://doi.org/10.2337/dc10-0368

Stade, M., Seyff, N., Baikenova, A., & Scherr, S. A. (2020). Towards a user feedback approach for smart homes: An explorative interview study. *2020 4th International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*, 5–10.

Stafford, T. F., & Farshadkah, S. (2020). A method for interpretively synthesizing qualitative research findings. *Communications of the Association for Information Systems*, *46*. https://doi.org/10.17705/1CAIS.04606

Stanik, C., & Maalej, W. (2019). Requirements intelligence with OpenReq analytics. *Proceedings of the IEEE International Conference on Requirements Engineering*, *2019-September*. https://doi.org/10.1109/RE.2019.00066

Stawarz, K., Preist, C., & Coyle, D. (2019). Use of Smartphone Apps, Social Media, and Web-Based Resources to Support Mental Health and Well-Being: Online Survey. *JMIR Mental Health*, *6*(7), e12546. https://doi.org/10.2196/12546

Sun, P. c., Cheng, H. K., & Finger, G. (2009). Critical functionalities of a successful e-learning system - An analysis from instructors' cognitive structure toward system usage. *Decision Support Systems*, *48*(1), 293–302. https://doi.org/10.1016/j.dss.2009.08.007

Sutanto, J., Palme, E., & Tan, C.-H. (2013). Addressing the Personalization-Privacy Paradox: An Empirical Assessment from a Field Experiment on Smartphone Users. *MIS Quarterly*, *37*(4), 1141–1164. http://www.misq.org

Sutcliffe, A. G., & Maiden, N. A. (1992). Analysing the novice analyst: cognitive models in software engineering. *International Journal of Man-Machine Studies*, *36*(5). https://doi.org/10.1016/0020-7373(92)90038-M

Takahashi, K., Takamura, H., & Okumura, M. (2005). Automatic occupation coding with combination of machine learning and hand-grafted rules. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3518 LNAI*(May 2005), 269–279. https://doi.org/10.1007/11430919{_}34

Tallyn, E., Fried, H., Gianni, R., Isard, A., & Speed, C. (2018). The Ethnobot: Gathering ethnographies in the age of IoT. *Conference on Human Factors in Computing Systems - Proceedings*, *2018-April*, 1–13. https://doi.org/10.1145/3173574.3174178

Tietz, T., Jäger, J., Waitelonis, J., & Sack, H. (2016). Semantic annotation and information visualization for blogposts with refer. *CEUR Workshop Proceedings*, *1704*, 28–40.

Tiwana, A., & Keil, M. (2006). Functionality risk in information systems development: An empirical investigation. *IEEE Transactions on Engineering Management*, *53*(3), 412–425. https://doi.org/10.1109/TEM.2006.878099

Tizard, J. (2019). Requirement mining in software product forums. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 428–433. https://doi.org/10.1109/RE.2019.00057

Tizard, J., Rietz, T., & Blincoe, K. (2020). Voice of the Users: A Demographic Study of Software Feedback Behaviour. *2020 IEEE 28th International Requirements Engineering Conference (RE)*, 55–65. https://doi.org/10.1109/RE48521.2020.00018

Tizard, J., Rietz, T., Liu, X., & Blincoe, K. (2021). Voice of the Users: An Extended Study of Software Feedback Engagement. *Forthcoming in Requirements Engineering Journal (REJ).*

Tizard, J., Wang, H., Yohannes, L., & Blincoe, K. (2019). Can a conversation paint a picture? mining requirements in software forums. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 17–27.

Toreini, P., Langner, M., & Maedche, A. (2020). Using eye-tracking for visual attention feedback. *Lecture Notes in Information Systems and Organisation*, *32*, 261–270. https://doi.org/10.1007/978-3-030-28144-1{_}29

Turel, O., Serenko, A., & Giles, P. (2011). Integrating technology addiction and use: An empirical investigation of online auction users. *MIS Quarterly: Management Information Systems*, *35*(4). https://doi.org/10.2307/41409972

Turk, A. M. (2012). Amazon mechanical turk. *Retrieved August*, *17*, 2012.

Tuunanen, T., & Rossi, M. (2003). Market driven requirements elicitation via critical success chains. *Proceedings of the IEEE International Conference on Requirements Engineering*, *2003-Janua*, 367–368. https://doi.org/10.1109/ICRE.2003.1232789

Tuunanen, T., & Rossi, M. (2004). Engineering a method for wide audience requirements elicitation and integrating it to software development. *2004 37th Annual Hawaii*

International Conference on System Sciences (HICSS'04), 1–10. https://doi.org/10.1109/HICSS.2004.1265420

Tuunanen, T., & Kuo, I. T. (2015). The effect of culture on requirements: A value-based view of prioritization. *European Journal of Information Systems*, *24*(3), 295–313. https://doi.org/10.1057/ejis.2014.29

Tuunanen, T., & Peffers, K. (2018). Population targeted requirements acquisition. *European Journal of Information Systems*, *27*(6), 686–711. https://doi.org/10.1080/0960085X.2018.1476015

Vaghefi, I., Lapointe, L., & Boudreau-Pinsonneault, C. (2017). A typology of user liability to IT addiction. *Information Systems Journal*, *27*(2), 125–169. https://doi.org/10.1111/isj.12098

Vagias, W. M. (2006). Likert-type scale response anchors. *Clemson International Institute for Tourism & Research Development*, 3–4.

Valkenier, M. (2020). Extracting high-quality end-user requirements via a chatbot elicitation assistant. https://dspace.library.uu.nl/handle/1874/394852

van den Hoven, J. (2017). Ethics for the Digital Age: Where Are the Moral Specs? *Informatics in the future* (pp. 65–76). Springer International Publishing. https://doi.org/10.1007/978-3-319-55735-9{_}6

Van Mechelen, M., Derboven, J., Laenen, A., Willems, B., Geerts, D., & Vanden Abeele, V. (2017). The GLID method: Moving from design features to underlying values in co-design. *International Journal of Human Computer Studies*, *97*, 116–128. https://doi.org/10.1016/j.ijhcs.2016.09.005

Vanden Abeele, V., Hauters, E., & Zaman, B. (2012). Increasing the reliability and validity of quantitative laddering data with LadderUX. *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, 2057. https://doi.org/10.1145/2212776.2223752

Villela, K., Hess, A., Koch, M., Falcao, R., Groen, E. C., Dorr, J., Valero, C. N., & Ebert, A. (2018). Towards ubiquitous RE: A perspective on requirements engineering in the era of digital transformation. *2018 IEEE 26th International Requirements Engineering Conference (RE'18)*, 205–216. https://doi.org/10.1109/RE.2018.00029

Wakefield, R. L., & Whitten, D. (2006). Mobile computing: A user study on hedonic/utilitarian mobile device usage. *European Journal of Information Systems*, *15*(3), 292–300. https://doi.org/10.1057/palgrave.ejis.3000619

Wang, C., Daneva, M., Sinderen, M., & Liang, P. (2019). A systematic mapping study on crowdsourced requirements engineering using user feedback. *Journal of Software: Evolution and Process*, 2199. https://doi.org/10.1002/smr.2199

Wang, C., & Lee, M. K. (2020). Why we cannot resist our smartphones: Investigating compulsive use of mobile sns from a stimulus-response-reinforcement perspective. *Journal of the Association for Information Systems*, *21*(1), 175–200. https://doi.org/10.17705/1jais.00596

Wang, W., & Benbasat, I. (2005). Trust in and Adoption of Online Recommendation Agents. *Journal of the Association for Information Systems*, *6*(3), 72–101. https://doi.org/10.1016/j.jsis.2007.12.002

Wiedemann, G. (2013). Opening up to big data: Computer-assisted analysis of textual data in social sciences. *Historical Social Research*, *38*(4), 332–358. https://doi.org/10.12759/hsr.38.2013.4.332-358

Wilhelms, M. P., Henkel, S., & Falk, T. (2017). To earn is not enough: A means-end analysis to uncover peer-providers' participation motives in peer-to-peer carsharing. *Technological Forecasting and Social Change*, *125*, 38–47. https://doi.org/10.1016/j.techfore.2017.03.030

Xiao, Z., Zhou, M. X., Liao, Q. V., Mark, G., Chi, C., Chen, W., & Yang, H. (2020). Tell Me About Yourself: Using an AI-Powered Chatbot to Conduct Conversational Surveys. *ACM Transactions on Computer-Human Interaction*, *1*(1). https://doi.org/10.1145/3381804

Xu, A., Rao, H., Dow, S. P., & Bailey, B. P. (2015). A classroom study of using crowd feedback in the iterative design process. *CSCW 2015 - Proceedings of the 2015 ACM International Conference on Computer-Supported Cooperative Work and Social Computing*. https://doi.org/10.1145/2675133.2675140

Yadav, S. B., Bravoco, R. R., Chatfield, A. T., & Rajkumar, T. M. (1988). Comparison of analysis techniques for information requirement determination. *Communications of the ACM*, *31*(9). https://doi.org/10.1145/48529.48533

Yamanaka, T., Noguchi, H., Yato, S., & Komiya, S. (2010). A proposal of a method to navigate interview-driven software requirements elicitation work. *WSEAS Transactions on Information Science and Applications*, *7*(6), 784–798.

Yan, J. L. S., McCracken, N., Zhou, S., & Crowston, K. (2014). Optimizing Features in Active Machine Learning for Complex Qualitative Content Analysis. *Proceedings of the ACL Workshop on Language Technologies and Computational Social Science*, 44–48. https://doi.org/10.3115/v1/w14-2513

Yang, H. W., & Chang, K. F. (2012). Combining means-end chain and fuzzy ANP to explore customers' decision process in selecting bundles. *International Journal of Information Management*, *32*(4), 381–395. https://doi.org/10.1016/j.ijinfomgt.2011.11.005

Yimam, S. M., Biemann, C., Eckart de Castilho, R., & Gurevych, I. (2014). Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, *1*(1), 91–96. https://doi.org/10.3115/v1/P14-5016

Yimam, S. M., Biemann, C., Majnaric, L., Šabanović, Š., & Holzinger, A. (2015). Interactive and Iterative Annotation for Biomedical Entity Recognition. *Proceeding of Brain Informatics and Health (BIH 2015)*, *9250*, 347–357. https://doi.org/10.1007/978-3-319-23344-4{_}34

Yousuf, M., & M.Asger, M. (2015). Comparison of Various Requirements Elicitation Techniques. *International Journal of Computer Applications*, *116*(4), 8–15. https://doi.org/10.5120/20322-2408

Zaman, B., Geurden, K., De Cock, R., De Schutter, B., & Vanden Abeele, V. (2014). Motivation profiles of online Poker players and the role of interface preferences: A laddering study among amateur and (semi-) professionals. *Computers in Human Behavior*, *39*, 154–164. https://doi.org/10.1016/j.chb.2014.07.009

Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In A. Aurum & C. Wohlin (Eds.), *Engineering and managing software requirements* (pp. 19–46). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-28244-0{_}2

List of Publications

Journal Publications

Tizard, J.; **Rietz, T**.; Liu, X.; Blincoe, K. 2021. Voice of the Users: An Extended Study of Software Feedback Engagement. *Requirements Engineering Journal (REJ).*

Journal Publications (Under Review)

Liu, X.; **Rietz, T.**; Sun, L.; Maedche, A. 2021. Repeated Use of Conversational or Graphical User Interfaces as Decision Aids: The Role of Task Completion Time and Rational Decision Style. Under review at *International Journal of Human-Computer Interaction.*

Rietz, T.; Maedche, A. 2021. Re-Evaluating User Values of Smartphones - A Wide Audience Qualitative Research Study Comparing Survey- and Chatbot-Based Laddering Interviews. Under review at *Information Systems Journal.*

Conference Proceedings (Published)

Haug, S.; **Rietz, T.**; Maedche, A. 2021. Accelerating Deductive Coding of Qualitative Data: An Experimental Study on the Applicability of Crowdsourcing. Forthcoming at *Mensch und Computer (MuC 2021)*, Ingolstadt, September 05-08, 2021.

Rietz, T.; Maedche, A. 2021. Cody: An AI-Based System to Semi-Automate Coding for Qualitative Research. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2021)*, Association for Computing Machinery (ACM). doi:10.1145/3411764.3445591.

Rietz, T.; Maedche, A. 2020. Towards the Design of an Interactive Machine Learning System for Qualitative Coding. International Conference on Information Systems, *ICIS 2020 - Making Digital Inclusive: Blending the Local and the Global*, December 13-16, 2020.

Rietz, T.; Toreini, P.; Maedche, A. 2020. Cody: An Interactive Machine Learning System for Qualitative Coding. *ACM Symposium on User Interface Software and Technology (UIST 2020)*, Online, October 20–23, 2020. doi:10.1145/3379350.3416195.

Tizard, J.; **Rietz, T.**; Blincoe, K. 2020. Voice of the Users: A Demographic Study of Software Feedback Behaviour [Distinguished Paper Award]. *Proceedings 28th IEEE International Requirements Engineering Conference: RE'20*; Zurich; Switzerland; 31 August 2020 through 4 September 2020. Ed.: T. Breaux, 55–65, Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/RE48521.2020.00018.

Rietz, T.; Schneider, F. 2020. We see we disagree: Insights from Designing a Cooperative Requirements Prioritization System. *Proceedings of the 28th European Conference on Information Systems (ECIS 2020)*, June 15-17, 2020.

Benke, I.; Guth, P.; **Rietz, T.**; Maedche, A. 2020. Discovering Core Modules of Platform-based Software Ecosystems for Non-Profit Sport Organizations. *Proceedings der 15. Internationalen Tagung Wirtschaftsinformatik (WI2020)*, Potsdam, March 08-11, 2020.

Rietz, T.; Maedche, A. 2019. LadderBot: A requirements self-elicitation system. *Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19)*, Jeju Island, ROK, September 23-27, 2019, 357–362, IEEE Computer Society. doi:10.1109/RE.2019.00045.

Rietz, T.; Benke, I.; Maedche, A. 2019. The Impact of Anthropomorphic and Functional Chatbot Design Features in Enterprise Collaboration Systems on User Acceptance. *Proceedings der 14. Internationale Tagung Wirtschaftsinformatik (WI2019)*, Siegen, February 23-27, 2019.

Conference Proceedings (Under Review)

Tizard, J.; **Rietz, T**.; Liu, X.; Blincoe, K. 2021. Voice of the Users: A study of software feedback differences between Germany and China. Under review at the CrowdRE workshop of the *29th IEEE International Requirements Engineering Conference (RE'21).*

Doctoral Consortium

Rietz, T. 2020. Designing a scalable AI-based requirements elicitation and analysis system. *Doctoral Consortium at the 28th European Conference on Information Systems (ECIS 2020)*, Marrakech, Morocco, June 15-17, 2020.

Rietz, T. 2019. Designing a conversational requirements elicitation system for end-users. *Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19)*, Jeju Island, ROK, September 23-27, 2019, 452–457, IEEE Computer Society. doi:10.1109/RE.2019.00061.

Eidesstattliche Versicherung

gemäß § 13 Abs. 2 Ziff. 3 der Promotionsordnung des Karlsruher Instituts für Technologie für die KIT-Fakultät für Wirtschaftswissenschaften

1. Bei der eingereichten Dissertation zu dem Thema *Designing AI-based Systems for Qualitative Data Collection and Analysis* handelt es sich um meine eigenständig erbrachte Leistung.
2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.
3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.
4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.
5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt. Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Karlsruhe, den *14.05.2021*

Tim Rietz