Incorporating UCD Into Software Organizations

A case study of work practices of a generic software implementation organization in India

Rebekka Ågesdatter Heggebø

Master Thesis
Informatics: Design, Use, Interaction
60 credits

Institute of informatics
The Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

June / 2020

Incorporating UCD Into Software Organizations

A case study of work practices of a generic software implementation organization in India

Rebekka Ågesdatter Heggebø

2020

Incorporating UCD into Software Organizations

http://www.duo.uio.no/

Abstract

This thesis investigates the application of the User Centered Design (UCD) methodology into the existing work practices of an organization that specializes in implementing generic software. Specifically, it focuses on the work practices of HISP India, who implement the software DHIS2, and the challenges that limit the feasibility of incorporating UCD into their practices.

The use of UCD processes has been shown to have several financial and business-related benefits. For example, reduced total time and cost of the development process, and increased end-user acceptance of the developed product. This thesis summarizes the challenges associated with following a user centered approach when implementing generic software. Generic software poses new challenges to the application of UCD as the design processes happen on two distinct levels, the generic (vendor) level and the implementation level. This thesis focuses on the processes that happen on the implementation level.

A major challenge when introducing UCD into software organizations is that many of the stakeholders within the organization do not see the benefits of following design methodologies. “Developer mindset” and “lack of knowledge” are frequently mentioned challenges that hinder this adoption. In addition to this, much of the work performed at the implementing organization is contract-based. This often results in the required resources to perform end-user involvement not being provided and can put restrictions on the implementers’ access to end-users.

The findings of this thesis are based on a two-year case study that has investigated the existing work practices established at HISP India. It presents the general workflow, challenges of incorporating UCD and where in the existing process UCD may be introduced. In this thesis I present increasing the organization’s knowledge about UCD as a possible aid in introducing this process. To achieve increased awareness and knowledge of UCD I describe possible solutions for how employees at HISP India can increase their knowledge about UCD, usability and other design related concepts. These solutions include DHIS2 Academies and mentor programs.

Keywords: User Centered Design, UCD, Design methodology, Generic software, Usability, User involvement, Design activities, Implementation-level design, DHIS2, HISP

Acknowledgments

I am forever grateful to my supervisor, Magnus Li, for his help and feedback on this thesis, which would not have been possible without his guidance.

I am indebted to HISP and HISP India for allowing me access to their work, taking time out for me, and warmly welcoming me to India. Without their cooperation and interest, the field work would not have been possible, and my stay in India would have been much less enjoyable.

I would like to thank the members of the DHIS2 Design Lab, and especially Elisabeth and Kristine, for our conversations, cooperation, and sparring.

Further, I would like to thank my fellow classmates in the group chat "Fokusgruppen" for always answering my questions and engaging in discussion.

Lastly, I would like to thank my parents, family, and Arne for supporting and loving me.

Table of Contents

List of Figures

List of Tables

Abbreviations

AMR	Antimicrobial Resistance
DHIS2	District Health Information Software
ERP	Enterprise Resource System
HISP	Health Information Systems Programme
HMIS	Health Management Information System
IFI	Institute of Informatics (at University of Oslo)
ISG	Implementation Specialist Group
IT	Information Technology
MoH	Ministry of Health
UCD	User-Centered Design
UiO	University of Oslo
UP	Uttar Pradesh
WHO	World Health Organization

Chapter 1 Introduction

This thesis investigates how the principles of User Centered Design (UCD) fit into the existing work practices of a software company in India that specializes in implementing generic health software. The aim of the research is to gain an understanding of the existing work processes, identify the challenges of involving users into these practices, and define areas of the existing practices that can be open for the introduction of elements of the UCD approach.

It has been suggested in research that user involvement can have a positive impact on the usability of a system and it can also lead to a quicker development process with a reduced need for expensive rework (Katsini et al., 2016; Otkjær et al.,2008). A study from 2014 identifies that only about 30% of software development companies perform activities to enable active user involvement (Ardito et al., 2014). This seems to be the case for software companies all over the world, with support from studies conducted in a few countries in Europe and in Ethiopia (e.g. Ardito et al., 2014; Katsini et al., 2016; Otkjær et al.,2008; Teka et al., 2017).

Studies conducted in varying geographical contexts find similar challenges reported by IT professionals regarding the obstacles they face when trying to involve users in the development process, with some local varieties. The most common universal challenges include “limited resources” and “lack of professionals”. We know from previous research that design methods need to be locally adapted to give value in different contexts (Baxter & Sommerville, 2011; Teka et al., 2017). This means that for UCD to be valuable beyond the context in which it was developed it must be adapted to fit the restraints and possibilities of the new context. This thesis discusses the challenges of introducing UCD in an organization that specializes in the implementation of generic health software in India. This includes two new contexts for a study of UCD, both the geographical and cultural aspects of India, and the technical aspect of the generic health software.

1.1 Motivation

For this thesis, I have been working with a software company located in northern India that specializes in the implementation of the generic, open-source software District Health Information Software 2 (DHIS2). DHIS2 is a web-based health information management system (HMIS) used in approximately 70 countries and is thus the largest HMIS in the world. DHIS2 is a platform used to report, analyze, and distribute data from numerous health programs (dhis2.org, n.d.). I have explored their work practices related to the project flow, investigating the activities that take place from a project’s beginning to the delivery of the final product. My motivation to carry out this study is threefold. Firstly, there is an internal aim at HISP India to adopt a more user-centered process, and a hope that this can decrease the amount of end-user training. Secondly, as the core HISP group at the University of Oslo generates a lot of research every year, this thesis can serve as a guide for future researchers interested in using a user-centered design approach within similar contexts. Lastly, I want to contribute to the growing body of literature on implementing UCD in software organizations, especially organizations that work with generic health software.

HISP India has expressed a wish to employ a more user-centered design approach to their development process and focus more on usability. A challenge with this is that the practitioners at HISP India have limited knowledge about practices surrounding the design discipline. The products developed by HISP India are either customizations of the existing applications in DHIS2 to fit their client's needs, or novel applications that provide additional functionality that extends the DHIS2 core. A large number of people are dependent on the usability of this software in order to be able to use it correctly and effectively. This can result in increased precision of health initiatives based on data from the platform. To be able to create meaningful systems that enable health workers to provide the best care for their patients, the creators of the software must rely on qualitative data from the health care providers when designing it (Li, 2019). This means that HISP India should progress towards a more user centered approach.

In my experience, the employees at HISP India have an interest in learning about design principles and related topics but they are too bound by strict schedules and overloaded with requirements from their clients. Another challenge is that the practitioners at HISP India have little to no interaction with actual end-users of the system and must establish a wish for this contact through their client. This is challenging because the involvement of users requires time and resources to be allocated to the project, which can be difficult to include in a contract that aims to be as cheap as possible for HISP India to win the bid. Contracts are an important aspect to consider when introducing UCD into software organizations as contract-based projects have been identified as a pain point that restricts the use of the UCD methodology in similar projects (Eshet & Bouwman, 2016).

My motivation is thus to examine HISP India's existing process to understand where a focus on user-centered design can be introduced. This knowledge can, in turn, be used to educate present and future employees at HISP India to adopt UCD principles into their routines. This could create a larger focus on UCD when negotiating contracts and implementing DHIS2.

1.2 Research Question

Involving the users in the development process offers the benefits of more usable systems, and also increases the probability of the system to succeed (Teka et al., 2017). This is not new knowledge, and Damodaran published already in 1996 that the participation of end-users in the development process could improve the quality of the developed product. It can also help the organization avoid expensive development of unnecessary features and result in greater approval of the system by the end-users (Damodaran, 1996). Even so, UCD is rarely used in software development processes (Ardito, Buono, Caivano, Costabile, Lanzilotti, et al., 2014). Generic software adds to the challenge of introducing UCD as it limits the flexibility of the implementation organization has, and often includes contract-based projects. To explore the presented problem, I ask the following research questions.

What are the challenges of introducing UCD into the existing work practices at a software company specializing in the implementation of generic software?

- *What parts of the existing work practices are open to the introduction of UCD?*

To be able to answer this question I have conducted a case study of the software company HISP India, which has given much insight into their work practices and possible areas of improvement. To understand their practices, I have been involved in ongoing projects, attended meetings, and working at their office in India, as well as conducting interviews that gave me a deeper understanding of their attitudes and opinions regarding their existing practices.

1.3 Chapter summary

Chapter 2 – Case Context

This chapter gives insight into the software, actors and projects that have been a part of my project. First, the chapter introduces the global movement HISP and the generic health management information system developed by them - DHIS2. Then, the actors that play a significant part in my research are also presented here. This includes the local HISP group, HISP India, and their client Alpha Consulting. Lastly, I describe the two main projects I have been involved in during my research.

Chapter 3 - Related Literature

This chapter first introduces the concept of User-Centered Design (UCD) and presents the stages of the process and four main principles related to *the process, roles, design activities,* and *user involvement.* Then, I describe the advantages and challenges of following a UCD process in the development processes in software organizations.

Chapter 4– Research Approach

This chapter provides the reasoning behind the chosen research methodology - case study. It also elaborates on the activities performed during this study and why these are appropriate for this study. The chapter follows a chronological narrative of the field trips, relevant data collection methods, and the methods used for analysis both during and after data collection are presented. Finally, the chapter introduces some challenges and limitations to the study.

Chapter 5 - Findings and Analysis

This chapter presents the results found during the study conducted. First, the chapter introduces the main roles and related skills of the employees and the general workflow of a project at HISP India. Then, I analyze the findings of the study and compare them to the ideal UCD process. Finally, the challenges HISP India have with introducing UCD into their process are introduced and the chapter reflects on how the employees want to learn about UCD.

Chapter 6 - Discussion

This chapter analyses the results found and discusses the findings in relation to the related literature. Specifically, the chapter includes a discussion surrounding the development process at HISP India compared to that of a traditional UCD approach.

Chapter 7 - Conclusion

This chapter summarizes the context of the project and the findings from the study.

Chapter 2 Case Context

The introduction described the general problem area and the specific research focus for my thesis: User Centered Design in organizations that implement generic health software. To gain insight into this problem area I conducted a case study of an organization in India and was included into a couple of their projects. First, I will give a definition to common words that have a specific meaning in this context. Then I describe the different actors that have played a part in my research, and the software that has been at the center of the study, DHIS2. Then, I will present the two projects I was a part of during my research with HISP India.

2.1 Terminology

This thesis uses terminology that has specific meaning in relation to the case. In this section I elaborate on the meaning I have assigned to these words throughout this thesis to avoid confusion. The words with explicit meanings are *develop, developer, implement, implementation,* and *implementer.* In this context, a developer is a person that adds to the functionality of generic software. This can be by developing new applications or HTML reports. When I use the term develop, in the context of something done by developers, I refer to the programming of novel solutions. Implementer is the role of the person who configures generic software without programming. To implement therefore refers to the process of adjusting generic software to fit a specific context. In the context of DHIS2, implementation refers to the process of adapting the software to fit the client's needs, and can be a result of implementing, developing, or a combination of both.

2.2 Actors

This section presents the different actors I have been interacted with as a part of my research. First, I present the global research community HISP and their software DHIS2. Then, I introduce two groups affiliated with HISP: The DHIS2 Design Lab and HISP India. Finally, I discuss one of HISP India's clients that I have been involved with during my research.

2.2.1 HISP

During the 1990s the University of Oslo started a research project in South Africa. During this time, they developed a software that would help the health sector in post-apartheid South Africa. HISP has since grown to become a global network of action that develops the generic software through a participatory approach (Braa et al., 2004). HISP aims to partner with ministries of health, universities and NGOs to empower local communities to improve the coverage and quality of health services (HISP, n.d.). To reach this goal HISP contributes to in-country capacity building, implementation support, research, and they also have a PhD program.

DHIS2

DHIS2 is a generic, open source software platform that supports a wide range of use cases. The software is highly configurable and allows for implementing organization to use the software to address local needs. The DHIS2 core is developed at the University of Oslo. This core team works to maintain and develop the core of the software which means apps and functionality that everyone gets. An example of what the software can be used for is disease surveillance. This works by collecting data about patients and analyze the data. This can enable prediction based on the data which might even prevent the next big outbreak. This software is, traditionally, mostly used in countries in the global South such as South Africa, Malawi, and Uganda. In March 2020 the COVID-19 outbreak officially reached Norway and we then saw that this software could be used to track COVID-19 patients in certain counties in Norway as well, so it can be configured to fit a variety of contexts and purposes.

HISP India

I have followed a local HISP node and their implementation process of the software DHIS2. The organization I studied is HISP India - Society for Health Information Systems Programme. In this thesis I will only refer to this organization as HISP India. HISP India is a part of the global HISP network and is a non-profit NGO that currently employs approximately six developers and 20 implementers. Their offices are located right outside of New Delhi in a city called Noida and specializes in implementing technical solutions for the health sector in India, Nepal, Bhutan, and a few other countries (HISP India, n.d.-b). The projects they are involved in include

DHIS2 Design Lab

The DHIS2 Design Lab is "an approach to addressing the usability and local relevance of generic gnterprise software" (Li, 2019). The actors involved in the lab fall into one of two categories: formal and informal members. The formal members consist of Li himself and a couple of master's students affiliated with the institute for informatics (IFI) at the University of Oslo (UiO). All the formal members of the lab conduct research within the fields of interaction design or programming and are in total approximately 20 people. The people the members of the design lab interact with while conducting research are seen as informal members. The lab's goal is to enhance the usability of the products created by the implementing organizations (*DHIS2 Design Lab - Department of Informatics*, n.d.).

2.2.2 Alpha Consulting

Alpha Consulting is a pseudonym to ensure the organization's anonymity. The organization is based in India, and works to enhance the wellbeing and health of people in India regardless of caste or religion, and they aim to achieve an equal public health sector of good quality. HISP India was contracted by Alpha Consulting to create a health information system (HIS) in the state Uttar Pradesh (UP). This is one of the largest projects HISP India is involved in right now.

2.3 Projects

While I conducted my research at HISP India I was not participate as a team member on any specific project, but I was partly involved in two different ongoing projects, "the UP project" and "the AMR project". This section present information about these projects that is relevant for this thesis.

2.3.1 The UP Project

During my first field trip to India I was moderately involved in the Uttar Pradesh (UP) project. Because HISP India is involved in implementing DHIS2 in the health sector, it is essential to understand the way the health system in India is structured. At the top is the national level which is the government and the Ministry of Health (MoH). This level has the national HMIS. Under the national level there are the various states and their governing bodies, such as the state Uttar Pradesh (UP) in northern India. District, block, and facility levels are geographical areas of decreasing size within a state. Each level reports to the level above.

Each state, including UP, are supposed to enter their routine data into the national HMIS. The state partnered up with Alpha Consulting who again contracted HISP India to achieve stable reporting from the health facilities in an integrated statewide HMIS called UPHMIS. The UPHMIS portal is integrated with the national HMIS. The state UP has a population of approximately 220 million people, so the UP in understandably one of the largest projects HISP India is working with as of today.

2.3.2 The AMR Project

The World's Health Organization's (WHO) general secretary has said that antimicrobial resistance (AMR) is "one of the most urgent health threats of our time" (*Antimicrobial Resistance*, n.d.). AMR is a huge, global challenge because it reduces the effectiveness of antibiotics when treating diseases (ibid.). The overuse of antibiotics in disease treatment for humans and in the livestock industry has led to bacteria evolving to survive the most common antibiotic treatments, leaving many drugs ineffective. This is a challenge because humans and animals are vulnerable to a set of diseases we are currently able to treat with these drugs. One way to slow the spread of AMR is by using surveillance systems that can locate outbreaks of resistant organisms. This way the health professionals can intervene and limit the reach of an outbreak. One of the projects I was a part of while studying HISP India's practices was the AMR Project, a project with the goal of creating a surveillance system for AMR.

Chapter 3 Related Literature

The previous chapter introduced the case context and described the various actors and technical aspects of my project. This thesis is about investigating the role of the User Centered Design (UCD) methodology in the context of software organizations who implement generic software, and this chapter introduces the theoretical positioning of the thesis. First, I present the concept of UCD and define the approach in terms of four main principles: *user involvement, roles, design activities,* and *the process.* Then, I introduce some advantages of UCD and user involvement in the development process. Then, I examine some challenges related to imcorporating UCD in software organizations, both perceived challenges from the eyes of the practitioners, and challenges related to the technical and the business aspects of the organizations. Finally, I summarize the main principles of UCD.

3.1 Defining User Centered Design

There is not one agreed-upon definition of what UCD is, so the first part of this chapter is a presentation of the definition I will lean on in my thesis. The term UCD was initially introduced by D. Norman and S. Draper in 1986. They emphasized that the "purpose of the technology is to serve the user", and the needs of the users must be considered when making the technology (Norman & Draper, 1986). Karat (1997) adds to the discussion by viewing UCD as an iterative process that has the goal of developing usable systems without strict guidelines as to how you should go about to accomplish that (Karat, 1997). The Interaction Design Foundation defines UCD as an iterative design process in which the users of the product are placed at the center. They emphasize that the goal is to involve the users in all phases of the development process so that in the end you have created a highly usable and applicable product (*What Is User Centered Design?*, n.d.). A common trait of the definitions is the use of the concept *usability*, which is defined by the International Standardization Organization (ISO) standard ISO 9241-11:2018 as follows:

> *" [The] extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"*

While all these definitions of UCD add to the overall understanding of the concept, Gulliksen et al. (2005) suggest that such general definitions will in practice turn UCD into a concept without any real meaning. They propose that the UCD approach must be more process-oriented and be integrated with the development process in such a way that usability and user-involvement is a united part of the process. In this chapter, I will present four central characteristics of UCD: *user involvement, roles, design activities,* and *the process.*

3.1.1 User involvement

All definitions of UCD have in common that the users of the system developed should be at the center of the process. Gulliksen et al. (2005) present two principles that address this aspect when using UCD. First, they build on Gould et al. (1997) and claim that all members of a project should know who the users are, what the use-context is, what motivates the users to perform certain tasks, etc., in order to help keep a focus on the user's needs instead of a technical focus. They suggest that all members of a project should have met with a real or potential end-user and that descriptions of users and scenarios should be posted around the workplace. Further, they suggest that active user involvement should be present throughout the development process (Gulliksen et al., 2003). They emphasize that a plan for the involvement of representative users should be ready at the beginning of a project. Maybe the most challenging principle of UCD is that a "User-centered attitude should always be established" (ibid.). This means that there needs to be an institutional change if an organization wants to follow the UCD process. There must be a change in attitudes and everyone involved in a project should be "committed to the importance of usability and user involvement" (ibid.).

Having a focus on and a willingness to involve users is the first step toward a UCD process, but there must be tools and methods in place to achieve this. To understand the users' needs, it is common to use techniques such as focus groups with representative users, interviews, observation of the users and their workflow, contextual inquiry, scenarios, and personas. These techniques help the project team to get a better understanding of what the users want and need from the system. User involvement can have several benefits, including user satisfaction, usable products and saving resources during the development process.

In order to build a useful product it is necessary to communicate with the future users of that product, and techniques such as document review and questionnaires are not sufficient by themselves (Damodaran, 1996). One reason for this is that sensitive information is more likely to be brought up at an interview, such information about problems the workspace or criticizing management (Otkjær et al.,2008). In this manner, inviting users to an interview gives the researchers both more information about the issues they were already aware of, and gives them insight on novel issues that the participants might not have been comfortable with disclosing in writing.

An interesting finding by Otkjær et al. (2008) was that there is a difference between organizations that do usability evaluating activities and organizations that do not when it comes to the attitudes they have towards involving users. They found that organizations that did not perform any design activities thought usability evaluation was unnecessary and stated this as the reason for not doing it. Contrary, the organizations that actually performed evaluating activities saw the value of performing these activities but emphasized that their lack of knowledge as the reason they did not conduct any activities related to user involvement or usability. They also found a difference in the positions the organizations took when it comes to who is responsible for conducting such activities. They found that non-evaluating organizations thought that it was the customers responsibility to involve the users, while evaluating organizations thought that the lack of participation was a result of the customer not seeing the business value (Otkjær et al.,2008).

3.1.2 Roles

In a software development project, there are usually several people involved. Gulliksen et al. (2005) argue that the development process should be carried out by a multidisciplinary team where the team members have the expertise that is needed to complete the various stages of the development process. They suggest that a team could consist of system architects, programmers, usability experts, and users. They interestingly regard the users as part of the team. There are many reasons why you should have a dedicated usability expert on your team when developing software. The Interaction Design Foundation claim that having a greater focus on UCD can help generate more revenue, decrease the project timeframes as well as creating better products for the users (*What Is User Centered Design?*, n.d.). To achieve this, Gulliksen et al. (2005) build on Kapor (1990) who insists that usability experts should be included in the development process early on, and continuously be a part of the process. The usability expert should then be the driver behind the UCD approach and make sure that it is followed throughout the project (Buur & Bødker, 2000). The usability expert should also have authorization to make decisions when a usability issue or something that can affect the user's interaction with the system (Gulliksen et al., 2003).

3.1.3 Design activities

In the principles that Gulliksen et al. (2005) propose they state that the development process should include dedicated and conscious design activities. They claim that the interaction design and the user interface of the product are a great part of what determines the success of the product. Keeping this in mind, the user interface of the product should be the result of deliberate design activities, and not a result of developers placing things in regards to the functional requirements (ibid.).

One example of a design activity that should be done throughout the design and development process is *prototyping.* Prototypes are a tool that can help designers communicate their ideas with the end-users, as well as other actors involved in the same project, such as developers. This helps create a shared language that is helpful when discussing and evaluating various solutions (Simonsen & Robertson, 2012). A key aspect of the UCD process is that prototyping should be done early on in the process and then continue throughout the process in order to support the creative process (Gulliksen et al. 2005).

There are different types of prototypes, ranging in complexity from quick sketches on paper to clickable, digital prototypes.

3.1.4 The Process

The intent of UCD is that is it a process that enables user-involvement, and key aspects of this process is that it is iterative and incremental (Gulliksen et al., 2003). When a new project is initiated it is near impossible to know for certain what the final product will be. The UCD process must thus support an evolutionary process where changes can be made continuously. As seen in Figure 3.1, the first step of the process after the beginning of the project is to understand the context of use. The remaining stages of the process include specifying user requirements, designing solutions, and evaluating the solutions against the user requirements. The UCD process is iterative so the stages do not have a strict order and the process is adapted as the project progresses. The double-headed arrows in Figure 3.1 imply that the design process can iterate over the parts of the process as required.

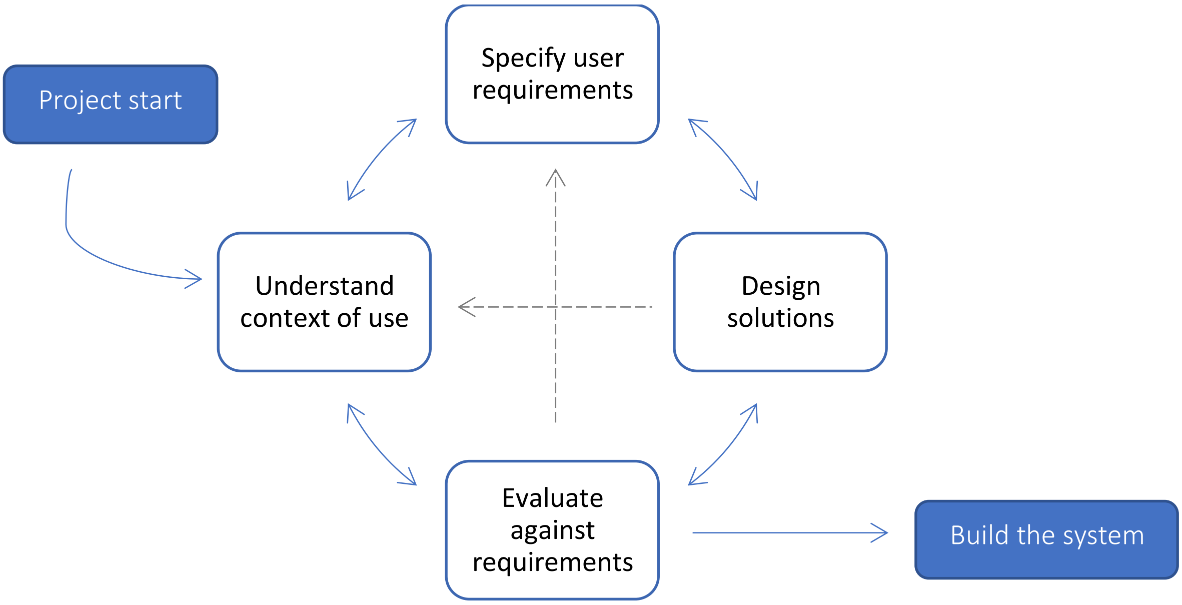
Project start
Specify user requirements
Understand context of use
Design solutions
Evaluate against requirements
Build the system

Figure 3.1 The ideal UCD process

While there are no restraints that dictate how long an iteration should last, is should contain the three steps: “1) proper analysis of the users’ needs and the context of use, 2) a design phase, a documented evaluation with concrete suggestions for modifications and 3) a redesign in accordance with the results of the evaluation” (Gulliksen et al., 2003). When an iteration is done the progress should be evaluated and adjustments should be made according to the feedback from the evaluation. This allows the project team to make informed decisions throughout the lifecycle of the project that can affect the usability of the finished product because they are able to detect what is and what is not working early on in the process.

When evaluating the product it is important to do so in the context in which it will be used. In the early stages of a project it can be valuable to determine what usability goals are relevant to the product and what criteria are success factors relating to the design of the product. When evaluating sketches and prototypes with the users the results should be evaluated up against these criteria (Gulliksen et al., 2003).

The design of a new product does not exist in isolation. It is important to remember the context in which the system will be used, and to keep in mind that the new product will affect the context when it is introduced (Gulliksen et al., 2003). When developing software to support work practices it is important to have a holistic view of all parts of the context such as what hardware is available and the social norms and practices that are in place (ibid.).

Perhaps the most important aspect of the UCD process is that it must be adapted to fit the local needs of the organization it is used in (Gulliksen et al., 2003). UCD is not a module that can be inserted, magically giving the organization a user-centered focus. The methods and tools used, and when and how they are used, must be incorporated into an organization in a way that is based on the organization's particular needs and constraints (ibid.).

3.1.5 Summarizing the Key Aspects of UCD

A summary of the central aspects of UCD that have been presented in the previous sections are described in table 3.1 below.

Table 3.1 Key Aspects of UCD

Aspect	**Description**
User involvement	All members of a development team should know who the future users of the system are, and there should be a plan in place from the beginning of the project for how and when to involve the users in the process.
Roles	Someone with expertise on usability-related subjects should be involved throughout the entire process and should have the authority to make decisions on matters that could affect the usability of the product
Design activities	Design activities such as prototyping should happen continuously throughout the development process.

Process	The design process should be iterative, and the evaluations done by users should be able to affect the developed product throughout the process.

3.2 Advantages of UCD

Overall there have been reported several benefits to following a UCD approach to the development of new products, such as improved quality of the final products and a reduces product failure (Otkjær et al.,2008). Some studies even suggest that a failure to consider both the social and technical aspects of a product can increase the risk of the final product being a failure (Baxter & Sommerville, 2011).

The benefits of effective user involvement during the design process has been well documented several times over and include improved quality of the product, and avoiding expensive features of the system that the user does not need due to a more accurate description of what the users need (Damodaran, 1996; Katsini et al., 2016; Otkjær et al.,2008; Teka et al., 2017). User involvement can be used as a tool to discover what changes to your system needs to be done before it becomes too complex and requires an expensive and time-consuming process in order to correct the misunderstandings the team had about user needs before the and developers have made the wrong product (Teka et al., 2017). The sooner the design and development team uncover problems they are less likely they are to be impossible to fix (Damodaran, 1996; Teka et al., 2017). This means that there is an economic advantage to involving users continuously throughout the process as this helps uncover any human and organizational issues (ibid.).

Already in 1996 Noyes et al. stated that earlier research supported that the involvement of users early in the process reduces the cost of development because you can detect issues early on (Noyes et al., 1996), and this is still applicable today (Katsini et al., 2016; Teka et al., 2017). Even so, cost is still one of the main reasons organizations give when asked why they do not involved users in their development process (ibid.). Interestingly, many organizations state that involving users is a resource demanding process, but were unable to express how it is resource demanding (Teka et al., 2017).

Many organizations also provide training of the users as a service. This is a time consuming and costly process that can be avoided to a certain extent if the product is usable. If the users are only involved as informants in the beginning and for acceptance testing at the final stages, you limit the chances of the users coming with useful insights that can increase the usability of a product, and thus lessen the resources needed to train them. There are many benefits of having a focus on usability throughout the process of developing a product, such as increased sales and reduced costs associated with training and user support (Kujala, 2003),which in turn would benefit the organization developing the product.

3.3 Challenges Related to Introducing UCD in Software Organizations

This section describes common challenges with introducing UCD in software organizations presented in existing literature. First, I present factors that limit an organization's motivation to use UCD or other usability enhancing methods. These factors are beliefs that are shared within software organizations across country lines and have, seemingly, been the same for the past ten years. Then, I present practical challenges software organizations face when trying to incorporate a UCD approach. I have sorted the challenges into the following categories: 1) Generic software, 2) The lack of cross-cultural considerations, and 3) Business complexity.

3.3.1 Perceived Challenges and Attitudes towards UCD and usability

When investigating the reasons why software organizations do not involve users or conduct any usability evaluations various studies have found similar reasons. The main reasons that have been identified are: "the developer mindset", "it is resource demanding", and "lack of professionals" (Ardito, Buono, Caivano, Costabile, & Lanzilotti, 2014; Katsini et al., 2016; Otkjær et al.,2008; Teka et al., 2017). There are also other reasons mentioned such as the lack of customer collaboration, lack of user motivation, users who do are not accepting of change, and some also state cultural influence as a reason to avoid usability enhancing activities (Teka et al., 2017).

Developer Mindset

In existing literature, the "developer mindset" is the most frequently mentioned reason for not involving users or doing any usability evaluation in the development process (Ardito, Buono, Caivano, Costabile, & Lanzilotti, 2014; Katsini et al., 2016; Otkjær et al.,2008; Teka et al., 2017). This is often a result of the developers not having knowledge about what these concepts entail, and thus they are unable to see the value (Otkjær et al.,2008). Often the responsibility is put on the users to learn about the technology and less responsibility is put on the IT professionals to learn about the formal and informal existing user practices (Damodaran, 1996). There are two main aspects to the developer mindset, 1) the developers are unfamiliar with the concepts of usability, and 2) the developers are not interested in that aspect of the software but instead focus on only the functionality. Otkjær et al. (2008) found that the concept of usability testing was often confused with the concepts of functionality testing, which are two tests that give very different kinds of feedback. The developers did often not accept the results of usability tests as they disagreed with the users' opinions and did not find their thoughts significant. Another aspect is that the developers often find that is more important to fix the bugs in the systems and generate "beautiful code" than it is to conduct usability tests (Otkjær et al.,2008). This shows that for an organization to have a more user centered approach to development there needs to be a change in the organizations views on, and knowledge about, these concepts.

Resource Demanding

After the developers' mindsets, the fact that these processes are resource demanding is also a common reason why an organization does not do any user involvement or usability evaluation (Ardito, Buono, Caivano, Costabile, & Lanzilotti, 2014; Katsini et al., 2016; Otkjær et al.,2008; Teka et al., 2017). Resources in this context means that these processes are time consuming and costly. Many organizations also state that the reason it is more expensive to do it is because they would need to have external assistance to be able to do these tasks. Some also mention that this service would add extra cost to the development process, which would have to be carried by the customer through an increased contract price. This is obviously not desirable to the customer (Otkjær et al.,2008). Some organizations stated that one problem with involving users is that sometimes evaluations done with users could uncover problems or complicated usability issues which are time-consuming and costly to fix (Teka et al., 2017). This worry is not only not supported by existing research, but contrary to these beliefs, early and frequent involvement of end-users will decrease the total project cost by reducing the risk of large and costly usability issues. This also comes to light when organizations were asked *how* exactly user involvement was resource demanding and many were unable to answer, which also should encourage the organizations to inform themselves about these topics (Otkjær et al.,2008; Teka et al., 2017). It is important to recognize that user involvement and other design activities are resource intensive as it demands that someone manage the activities, but that the overall cost of the project can be cheaper when such activities are done.

Lack of professionals

Lack of knowledge about UCD is likely the reason for the impressions that user centered design is a resource demanding process. Teka et al. (2017) conducted a study in Ethiopia in 2017 and found that public universities did not include have any courses about usability or human computer interaction until late 2014. In Denmark, where Otkjær et al. conducted their study in 2008, there has traditionally been more focus on user involvement in software development, and this is also where the discipline of participatory design originated in the 1970s. Even so, Otkjær et al. and Teka et al. still share many of the same findings regarding obstacles surrounding user involvement in software organizations. Otkjær et al. also report that the software organizations they studied reported that they found it difficult to conduct tests and thus avoided it as they did not feel like they got any value from it. A few organizations also mentioned that it was a challenge to convince their customers to participate in the design process as the customers did not have the necessary knowledge to appreciate why they should be involved (Otkjær et al.,2008). Damodaran (1994) states that it is essential that a user centered approach must be incorporated in a top-down fashion and in that way be embedded in the organization for it to work. To involve users, you need resources and the authority to make changes based on user feedback. It is therefore important that top level management is on-board and provide this authority. Otkjær et al. (2008) also found that a high turn-over rate of developers poorly affected the general usability work as this limits the transference of gained knowledge.

3.3.2 Generic software

The concept of generic software has many names – *generic, off-the-shelf, packaged, ERP software,* etc.*,* which all describe software that can be adapted and used in various contexts (Li, 2019). The generic software has a several different actors involved in the development process. The most central actor is the organization that leads the development of the generic parts of the software. These actors are often called the vendors. The vendors produce the main product and focus on design on a generic level. In the case discussed in this thesis the vendor is the DHIS2 core developer team located in Oslo, Norway.

On the generic level, the designers and developers make resources intended to be used by many others, such as generic data collection and analytics applications. Combining generic software and a UCD process can therefore be a challenge. Generic software is created from the "top" and made to fit into several contexts and serve multiple purposes. The UCD process on the other hand, focuses on understanding the specific context the software will be used in. On the generic level it is hugely challenging to follow a UCD approach, since the designer cannot consider specific contexts and users. This in turn has been shown to lead to reduced usability in the implemented local instances of the software (Li, 2019).

Before the product from the vendors reaches the end-users, someone must make adequate adaptions to the software to make it suitable for the context it will be used in. This is done by implementation-specialist groups (ISGs). The main roles of the ISG are to help their clients implement the generic software. They can also develop their own apps to extend the functionality of the generic software to make it suitable for their clients (Li, 2019). The ISGs utilize the resources made available to them by the vendors of the software. For this study I have studied the work practices of the ISG HISP India.

At the implementation level, implementors are largely restrained by the available interfaces and functionalities that exist, and sometimes do not see conducting usability evaluations as their responsibility, since they are merely configuring the software for their own context (Katsini et al., 2016). The reasoning behind this is that the products are already in use, and the software organization only configures the product to fit their needs. This is a challenge for organizations that mostly implement generic software, and it is suggested that generic software is a hindrance in a process with a goal of making products that are usable by the end-users (Li, 2019).

The challenge with following a UCD approach on the implementation level is that they get a package that contains much pre-defined interfaces and functionality and their work mainly revolves around adjusting it to fit their context. This problem is not much addressed in literature, although some literature suggest that the traditional UCD methods do not support projects that are not developed for a single use-case and context (Li, 2019). On the generic level it is very challenging to introduce a UCD approach as the designer cannot consider specific contexts and users, which results in reduced usability in the local instances of the software (ibid.).

There are two main challenges with following a UCD approach when implementing generic software: 1) Generic software is made to fit into many contexts and serve several purposes, and 2) The generic nature of the software imposes a top-down design process when implementing it due to the predefined interfaces and functionality (Li, 2019).

3.3.3 Business Complexity

Few studies recognize the business complexity when studying the use of UCD in software organizations. The main challenges that software organizations need to work with is 1) Contract-based projects, and 2) Clients who restrict access to users and resources (Eshet & Bouwman, 2016). Contract-based projects are challenging because the contract dictates the type of work to be done and can limit the resources dedicated to a project. The degree to which users are involved, and the availability of resources required for this, depend on the culture and work practices of both the software organization and the client (Damodaran, 1996). Given these challenges, the practitioners who try to understand the user's needs and the context of use often rely on third order understanding of these aspects as they do not have easy access to the end-users and rely on information given to them by their client. Therefore, in the case of contract-based projects, it is encouraged to establish long-lasting relationships between the software organization and the clients (Eshet & Bouwman, 2016). It is also beneficial to discuss the advantages and possibilities of UCD at an early stage in the project (Damodaran, 1996) in order to get a UCD focus established in the contract.

3.3.4 Cultural Considerations

There are many methods to use when trying to incorporate a UCD process, such as interviews, prototyping, personas etc. It is well-known that the interface of the software must be adapted to fit the needs of the users in order to be usable, but there are also indications that the methods used to gain insight into user needs also need to be modified to the context they are being used in (Teka et al, 2017).

Because user involvement is traditionally a western concept, many of the methods used by researchers are developed in a western context. This means that one cannot take for granted that a method developed for the western context will work in the setting of a developing country (ibid.). Teka et al. (2017) confirm this through their study in Ethiopia where they found that the method of usability evaluation was difficult to conduct because the intended end-users did not have the necessary IT knowledge.

3.3.5 Summarizing the Common Challenges of Introducing UCD

The main challenges associated with introducing UCD intro software organizations that have been introduced are summarized in Table 3.2.

Table 3.2 Summary of Challenges with introducing UCD

Challenge	Description
Developer mindset	Many developers are not familiar with design processes and usability and would rather focus on the technical aspects of the product.
Resource demanding	Resources like time and money are common obstacles organization point to as the reason they do not follow a design methodology such as UCD. An addition to the resources, many organizations also need eternal support because they do not have the needed expertise themselves.
Lack of professionals	Many organizations do not have employees with expertise on design processes. In some countries there is also the added obstacle of lacking formal education options within the field of systems design.
Generic software	Generic software is made to fit into several contexts while UCD focuses on analyzing one specific context and developing the product based on the context. Another challenge with generic software is the default top-down approach as a result of the dynamic between the manufacturers of the generic software and the implementers.
Business Complexity	Contract-based projects and clients who restrict access to end-users are challenging aspects of the business side of software development.
Cultural Considerations	Methods for user involvement and design activities must be adjusted to the context it will be used in.

Chapter 4 Research Approach

The DHIS2 Design Lab had already established a relationship with HISP India, which allowed me to do my research project with them. The DHIS2 Design Lab aims to achieve socio-technical design which focuses on both the technical aspects of the solutions, but also the social structures that surround it. One of HISP India's clients suggested that they kept training the same users on the same things over and over, so I wanted to explore how a UCD process could fit into HISP India's work practices, and I wanted to use this approach in a project. It has been suggested many times that the process of involving users bring significant benefits to a development process, but research is showing a lack of user involvement in practice (see chapter 2). I find it very interesting that there is such a noticeable gap between research and practice and wanted to investigate this further. The question that guided my research is:

What are the challenges of introducing UCD into the existing work practices at a software company specializing in the implementation of generic software?

- *What parts of the existing work practices are open to adopting the principles of UCD?*

To answer this, I needed to explore the implementation process of an organization that specializes in the configuration of generic software, and I needed to get familiar with the UCD process. While learning more in depth about the UCD process could be done through a literature review, learning about real life work practices required me to observe and interact more with the people who perform this type of work. In this thesis, I explore how the work practices at a software organization might be introduced to UCD principles. To do so I have conducted an interpretative case study with HISP India by following two of their projects and the day-to-day activities the employees perform.

This chapter aims to give a detailed description of the approach I had when investigating this question and area of interest. The rest of the chapter is structured as follows. First, I give an insight into my reflexivity as I present my position and philosophical viewpoints. Then, I justify my choice of research methodology and present the different stages of my research project and my goals at each stage. Then, I go through the data collection methods I have utilized to get an understanding of the concepts I study. Finally, I describe the methods I have used to analyze my findings, both during my stays in India and after.

4.1 Philosophical views

When conducting a research study, it is important to be aware of what assumptions you as a researcher have of the world. I subscribe to the interpretive paradigm because I think that in order to be able to understand how the principles of UCD fit into HISP India's practices I must understand the meanings that the employees assign to these principles (Klein & Myers, 1999). I also believe that the only way I can get access to reality in this context is through social constructions such as language, consciousness, and shared meanings. So, to understand something, I must understand the context in which it happens from the perspective of the people I aim to understand. In my case this means that I must understand the work practices from the viewpoint of the employees at an organization that implements generic software. A key concept of the interpretivist paradigm is the belief that intersubjectivity is essential to understanding. This is the notion that the shared meanings constructed by the interaction between humans is needed to gain knowledge (Orlikowski & Baroudi, 1991). These are beliefs that I share with other researcher who also conduct research that relies on qualitative data and is the reason I have chosen to focus on qualitative methods for data collection and analysis (Walsham, 2006).

4.2 Methodology

Choosing a methodology to guide my research has not been a straightforward process in my project as the seemingly most suitable approach has changed as I have adapted the focus of my thesis as the project unfolded. This section starts by giving an overview of the different stages of my project, and then I go into detail about the methodology I have chosen for my research – Case study.

4.2.1 Finding a Suitable Research Approach

When I started my research, before my first field trip, I believed that HISP India involved users to a much greater extent than they actually do. At this point, I focused on the participation and empowerment of the end-users in accordance with the Scandinavian tradition of Participatory Design (PD) (Simonsen & Robertson, 2012). During this time, my plan seemed to fit the theoretical aspects of the Action Research (AR) methodology because I wanted to take part in the work practices and be involved in the change of some aspects for the process.

On my first trip to India, I went with my supervisor and five other master's students working on separate projects and stayed for four weeks. We spend the working hours at HISP India's offices in Noida, having meetings, presentations, workshops, and group interviews. Through these activities, I learned much about HISP India as an organization and the DHIS2 software. At this time, I realized that user involvement happened on such a small scale that it could be a challenge to introduce PD given the time frame of my thesis, especially through an AR approach.

During my first visit in India, I learned about the context in which the software DHIS2 is used and who it is used by. So, when I returned to Norway, I shifted the research focus of my project. The context had been clarified for me in such a way that I was able to gradually shift the focus of my research before my second field trip. Between the two trips to India, I continued reading relevant literature on UCD, PD in various contexts, and how to use different techniques such as card sort and personas. My focus moved from the participatory approach to a user centered approach. The reason for this is that PD focuses on the empowerment of users, democratic processes, and aims to develop products in a "bottom-up" approach (Simonsen & Robertson, 2012). Having a user centered approach, on the other hand, allowed me to be more pragmatic and usability-oriented, which was more in line with the observed behaviors at HISP India. I chose to focus on introducing design processes, tools, and techniques into the existing work processes.

During my first field trip to India I was a part of the "UP-project", a large project that aims to integrate a health management portal across the state Uttar Pradesh (UP) and connect this to the national HMIS. During my stay I realized that this project did not have room for me so while I was in Norway I was partly included in a different project, the "AMR-project". The AMR project is a disease surveillance system that tracks outbreaks of antimicrobial resistance. In relation to this project, I evaluated apps that two fellow master's students had made for this project, and attended a workshop held at the University of Oslo where various actors that are engaged in the field of AMR attended. This included IT people, medical doctors, and researchers.

When I arrived India for the second, and last, time I hoped to be included into the AMR project - at least to some extent. I realized early that the AR methodology would be very challenging as any footing I had in HISP India seemed to have disappeared since my last trip. This might be a result of several things, but I suspect that mainly two things attributed to this: 1) The people I had spent time getting to know during my first stay were no longer at HISP India, and 2) I was not officially a part of any project nor had it been officially organized that I was a part of the AMR project. So early on during my second stay I realized that the case study methodology was the approach that could best help me answer my research question(s). Figure 4.1 shows an overview of the different stages of my projects, the research focus I had, and the methodology I tried to follow.

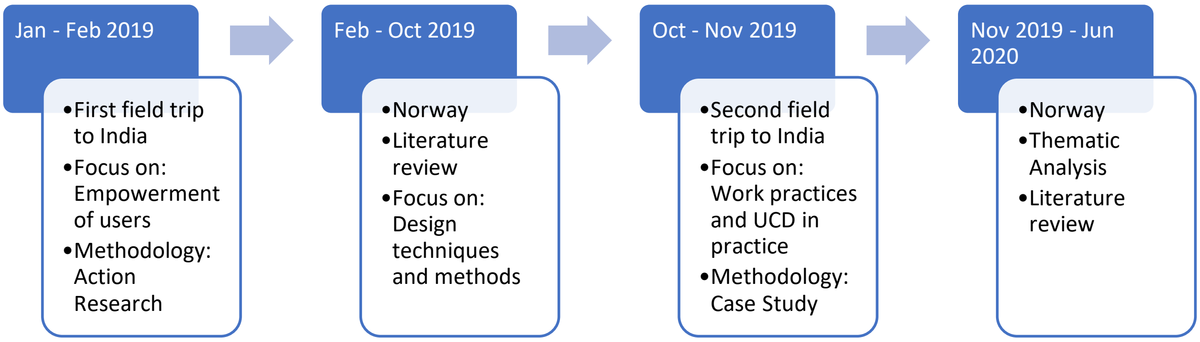
Jan - Feb 2019
•First field trip to India
•Focus on: Empowerment of users
•Methodology: Action Research
Feb - Oct 2019
•Norway
•Literature review
•Focus on: Design techniques and methods
Oct - Nov 2019
•Second field trip to India
•Focus on: Work practices and UCD in practice
•Methodology: Case Study
Nov 2019 - Jun 2020
•Norway
•Thematic Analysis
•Literature review

Figure 4.1 Overview of the different stages of the research project

4.2.2 Case Study

Case studies are arguable the most common research methodology in information systems research (Orlikowski & Baroudi, 1991). Yin, Stake, and Merriam are arguably the three most influential voices when it comes to classification and descriptions of what a case study is (Yazan, 2015). Yin describes a more positivist approach to defining and conducting case studies, so this definition is not fitting given the interpretive approach I have employed to understand the practices at HISP India (ibid.). Merriam has a similar approach to case studies as the one I have but argues for the importance of conducting a literature review that yields a framework that guides the formation of the research question(s) (ibid). Given the fluctuating focus of my study it was impossible to conduct a proper literature review before the data-gathering started. Stake (1995) debates that the researcher should be able to have a flexible research design which allows major changes even after the research has begun (Stake, 1995, as cited in Yazan, 2015). Stake embraces the term "progressive focusing" introduced by Parlett and Hamilton (1972) which entails that you do not know beforehand what is going to happen, and you must, therefore, be prepared to redefine your research as you go along (Yazan, 2015). I find this very fitting for my research as I started out with a vague idea of what I was looking at during my first trip. This idea became more defined while I in Norway between field trips and was clear when I returned to India for the second time. I also encountered obstacles I would not have been able to overcome without the opportunity to adjust my focus according to the situation, such as realizing the end-users were never actually in contact with HISP India. Stake's definition of a case study emphasizes the importance of the researcher's awareness of how they understand information. He argues that this affects how you conduct your study. This view on knowledge is connected to the interpretative paradigm, as is mine, so I found this description the most fitting when characterizing my study.

Stake (1995) does not define what a case is, but instead agrees with Smith's (1978) definition which states that a case should be viewed as "a bounded system" that you observe "as an object rather than a process" (Yazan, 2015). However, Stake does give four characteristics that he believes describe qualitative case studies: holistic, empirical, interpretative, and emphatic (ibid.). A holistic approach will entail that the researcher is aware that the smaller parts of a system are interconnected, and that they can only be understood if the researcher considers the complete system. This means that the link between the phenomenon and the surrounding context must be taken into consideration. In my case I found it challenging to understand some of the decisions and approaches made by the people I studied when I only examined their actions within the office. It was then beneficial for me to try to see the bigger picture and adopt a more holistic approach. The DHIS2 Design Lab helped me adopt a more holistic approach as I could discuss with people with the same status, being "outsiders" such as myself, but with a different theoretical background.

The characteristic that states that the case study should be empirical also fits my case as it is based on empirical data that I have gathered during observations in the field. As Stake's epistemological views are constructionist, he believes that knowledge is constructed. An interpretive approach to data gathering is therefore essential. This means that to gather data the researcher must establish a relationship between themselves and the subject they are studying. This is also compatible with my approach, as I spent much time trying to get to know the employees I studied, and much of my knowledge comes from casual conversations I had with them during lunch, having dinner together, or just in passing at the office. Finally, Stake (1995) emphasizes that the researcher must be empathic when doing a case study. This means that the researcher must try to understand the subject's experiences from their perspective to gain knowledge about it. This was a challenge for me given that I studied both an unfamiliar context and culture, but again the DHIS2 Design lab helped me analyze my findings through discussions and meetings. The people met at HISP India were also willing explain in more detail if I found something confusing.

4.3 Practical Aspects of the Project

This section gives an overview of the actors involved in my project and the overall information about the activities I have conducted through the course of my thesis.

4.3.1 Actors and Participants

In the remaining part of my thesis I will refer to a few different actors. In this section I present the actors and activities that are central to my research to make it more straightforward to follow along with the events described. The actors involved can be divided into three main categories: people from the University of Oslo (Table 4.1), people from HISP India (Table 4.2), and people I interacted with on field trips in India (Table 4.3).

Table 4.1 University of Oslo

My fellow master's students. Often referred to as "we".	This refers to the group of master's students who were some of the first members of the DHIS2 Design Lab and who went to India during the same periods as I did.
Supervisor	This refers to my supervisor, a PhD candidate at UiO
Core team	This refers to the team (mainly) located at UiO who are responsible for developing the core functionality of the software DHIS2.
DHIS2 Design Lab	This refers to the research community that is the DHIS2 Design Lab which is described in more detail in chapter 2.2.1

Table 4.2 HISP India

Developers	This refers to the group of approximately 6 developers who are employed by HISP India to do development and have a formal education within computer science/informatics.
Implementers	Implementers refers to the people employed by HISP India to do public health analysis and design. This group mainly work with understanding and gathering requirements and adjusting the configurable parts of DHIS2 to fit the clients' needs without development, which is referred to as implementation in this thesis.
Interns	During my first stay in India, HISP India had employed five interns who studied biology to work as implementers.
Newly employed implementers	This refers to the two interns who were hired after their internships (and studies) were done. During my first trip they were interns, and they had worked as implementers for a few months when I came back for the second time.

Table 4.3 Actors from the field

Alpha consulting	One of HISP India's largest contracts is with this organization. HISP India works with this organization to develop the UPHMIS (see chapter 2.2.2 and 2.3.1). For HISP India to visit the field and meet users this needs to be facilitated by Alpha Consulting.
Data entry operator (DEO)	A data entry operator is a person whose job it is to enter data into a system. DEOs are exposed to DHIS2 as their job is to enter data into this system as well and they are considered end-users of the system. DEOs are a part of this thesis because they are one of the user groups we met as a part of the UP and AMR projects.
Monitoring and Evaluation (M&E) Officers	Alpha Consulting employs M&E officers who aim to increase the performance of the district they are a part of. These are also users of DHIS2.

4.3.2 Methods for Data Collection

During both my visits I used the same techniques for data collection, so my experiences during the first visit gave me knowledge that guided my actions when I came back the second time. Stake (1995) suggests interview, observation, and document analysis as suitable methods for data collection, which is also what I have done. He also emphasizes that all data collection methods should be qualitative when conducting a case study (Yazan, 2015). In addition to Stake's recommended methods, I have been involved in the facilitation of workshops where we used more tangible methods, such as paper prototyping.

Table 4.4 gives an overview of the various activities conducted during the different stages of my project. While in India I focused on qualitative methods such as interviews, observations, and workshops. In addition, I was also given access to a few documents while in HISP India's offices and used e-mail correspondence to gain more knowledge about the context I was in. I was part of two workshops between the two fieldtrips. During my second field trip I used many of the same methods as I did during my first stay in India. This section describes these methods and the reason I chose to use them.

Table 4.4 Overview of Activities

Period	Location	Activities	Main areas of learning
January 2019	India	6 Interviews – Groups and individual	The projects at HISP India and the HISP/DHIS2 ecosystem.
		Observation – Passive and Participatory	Relationships between HISP India and their clients, general practices at HISP India and the process of getting feedback from health facilities.
		2 Workshop	HISP India's attitudes towards using design techniques such as prototyping.
February - October 2019	Norway	Design Systems workshop with ABB	Design systems in large, international organizations.
		Method workshop	What to keep in mind when facilitating workshops and reflections around suitable design activities.
		AMR workshop	The context the AMR project was in.
		Literature review	General principles of UCD and how to use various design techniques.
October - December 2019	India	6 Interviews – Groups and individual	Practices at HISP India related to implementation and development.

		Observation – Passive and Participatory	Routines at HISP India's offices and behaviors during workshops and fieldtrips.
		Workshop	Attitudes toward usability, prototyping and user involvement. The workshop also gave insight into the existing work practices at HISP India and how they would prefer to learn about new concepts.
December 2019 - June 2020	Norway	Thematic analysis	Main categories of findings related to UCD: process, roles, design activities and user involvement.

Interviews

Interviews are one of the most common methods used in qualitative research, and often used in combination with observations to get a more nuanced view of a situation (Edwards & Holland, 2013). For my research I have used mainly semi-structured interviews as these have given both me and the interviewee the opportunity to follow the flow of the conversation. The semi-structured nature of the interviews also lets the participant talk freely and use concepts and ideas they are familiar with (ibid.).

During my first trip to India I conducted six formal interviews with interns, implementers, and developers. This saved time for the employees at HISP India and made them more willing to answer my questions. During my second stay I also conducted six interviews, this time with both newly employed and experience implementers as well as two developers. Table 4.5 shows an overview of all the "formal" interviews I have conducted that are relevant for this study, and the main learnings from each interview. In addition to these, I had several interviews that are hard to tally as many were just a few questions over lunch or after a meeting and was more like a normal conversation.

Before the interviews I conducted alone I planned 3-4 questions I wanted to ask and focused on topics I was interested in learning more about and tried to steer the conversation in that direction if that felt natural. I kept these interviews very open and almost un-structured. Most of these interviews were conducted during the first fieldtrip when my goal was to gain as much insight into the context as possible. The interviews I conducted in collaboration with the other master's student were also semi-structured. We asked questions like "Can you tell us about a project you have recently been involved in?" and "What challenges do you face during projects". We varied the questions a little to fit the interviewee's role. The main topics we asked about were communication, challenges, and project flow.

The goal of conducting semi-structured interviews was to gain more detailed information about specific topics that I was interested in, such as work practices and attitudes towards user involvement, prototyping and similar concepts.

Table 4.5 Overview of formally conducted interviews

Interviewee's role	Interviewer(s)	Fieldtrip	Main areas of learning
3 Interns	Me	1	What it is like to be new at HISP India and how they experienced the group work at the workshop. The interview also gave insight into the employees' background.
Implementer	Me	1	How the pivot table application in DHIS2 works.
Developer	Me	1	Who the users of the pivot table application are and how DHIS2 works.
Developer who had given his notice of resignation	Me and another master's student	1	General reflections about the culture at HISP India.
Implementer	Me	1	The hierarchy in the office, reflections regarding workshops and the role of implementers.
Developer	Me	1	The practices surrounding development such as functionality and usability testing.
2 Newly employed implementers	Me and another master's student	2	The transition from intern to implementer.
Implementer	Me and another master's student	2	General flow of a project from start to finish and challenges faced.
Implementer	Me and another master's student	2	General flow of a project from start to finish and challenges faced.
Implementer	Me and another master's student	2	General flow of a project from start to finish and challenges faced.
Developer	Me and another master's student	2	The role of developers in HISP India's projects, challenges faced and modes of communication with implementers.
Developer	Me	2	Reflections surrounding resources that can be used to learn about UCD/usability etc.

Observations

Much of the knowledge I gained during my stays in India was through passive and participatory observations. Much of the participation fits Zahle's (2019) description of weaker participation as we sat in the office and worked alongside the employees at HISP India and took notes of things we observed. We were invited to do field work at HISP India through an existing collaboration with UiO, and I thus avoided the common challenge with case studies which is that it is difficult to get access to the field (Crang & Cook, 2007). Since HISP India ended up as my unit of analysis, I was very fortunate to be given this level of access.

Workshops

Table 4.6 shows an overview of all workshops I was involved in during my project and what my role in the workshop was. The different workshops served as means to reach different goals. Workshops 1 and 5 gave me insight into the practices and attitudes toward design processes at HISP India and thus helped me answer my research question. While not helping me answer my research question directly, workshops 2,3 and 4 helped me get a better understanding of both design techniques and processes and the context I was working in. More details about the workshops are presented in chapter 4.4.

Table 4.6 Overview of workshops

Activity	No. participants	My role	Location	Date(s)
Workshop 1 - part 1: Reporting Progress	14	Co-facilitator with another master's student	Noida, India	Feb 4th, 2019
Workshop 1 - part 2: Reporting Progress	10	Co-facilitator with another master's student	Noida, India	Feb 6th, 2019
Workshop 2: Design Systems with ABB	10	Participant	University of Oslo, Norway	Mar 27th, 2019
Workshop 3: Method testing	6	Facilitator	University of Oslo, Norway	Mar 29th, 2019
Workshop 4: AMR	Approx. 25	Participant	University of Oslo, Norway	Apr 11th, 2019
Workshop 5: Design of locally meaningful technology	Approx. 16	Co-facilitator with my supervisor	New Delhi, India	Nov 11th - 15th, 2019

4.4 Field trips

This section goes in more detail about the process briefly described in chapter 4.2.1, with a focus on the empirical activities during my two field trips to India and my main learnings from the different stages. I go through the three main stages of my process: 1) First field trip to India, 2) Between fieldtrips, and 3) Second fieldtrip to India.

Figure 4.2 to the right shows a map of India with markings showing where I was. The two blue markers indicate the HISP India office (left) and the Alpha Consulting office (right). The three red markers show the health facilities we visited. The top one is related to the AMR project and the two to the right are related to the UP project.

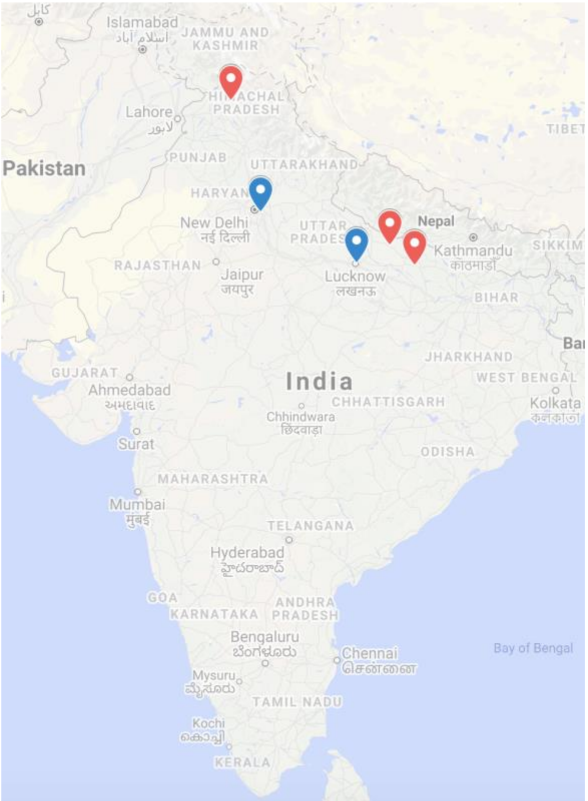

Figure 4.2 Map of India with markings representing the locations visited during the two fieldtrips

4.4.1 The First Fieldtrip

My first trip to India happened quickly after I had decided on a research field for my thesis, and I had not focused on a specific problem I wanted to investigate. I was thus very open minded at the beginning of the first stay in India, and I was open for almost any challenge HISP India thought I could help solve. My goal during this first trip was to learn about India, the HISP network and DHIS2, as well as the work practices at HISP India. I also aimed to establish relationships to the employees at HISP India that could help me when I returned for my second field trip.

Introduction to the projects

During the first week we spent a lot of time in meetings and wathcing presentations on the work HISP India does and their current projects. We got a lot of information about the UP project as it was decided that we should partake in that project. To get more information about the software DHIS2, which I had limited knowledge about, I had a few interviews with both implementers and a developer. I had a "counterpart" at HISP India who was also very helpful when I had questions. I was "assigned" one of the problems they had within the project. They wanted me to fix the *pivot table* app, so that was what my focus during this time.

Alpha Consulting and Field Visits

During this trip, my supervisor, three fellow master's students and I went to a different city, Lucknow, to meet with an organization that HISP is contracted by, Alpha Consulting, to participate in the UP project. They arranged for us to visit two health facilities, one in Balrampur and one in Khalilabad, both rural areas in the state Uttar Pradesh. On our visit to the first facility in Balrampur, we had a meeting with many representatives from the health facility, a high-up employee from HISP India and two representatives from Alpha Consulting in addition to two of my fellow master's students and me. This meeting lasted for approximately 40 minutes and was mainly in Hindi.

During the field trip to Khalilabad, we met with a data entry operator (DEO) and a district monitoring and evaluation officer (DM&E) who worked at a hospital. We interviewed them and observed how they worked. One master's student also tried to have the DM&E draw a low fidelity prototype of what he wanted. He drew one line and then went back to talking. I also tried to talk to the DEO one-on-one but relied on a translator as he did not speak English, so the conversation was kept short.

These visits to the field gave me insight into the processes and practices that go on in the field. I got to observe some of the intended end-users of HISP India's products and the equipment they used to do their jobs. In other words, I observed how the first step of UCD, "understand the contest of use", could be completed. I also observed the attitudes from both HISP India and Alpha Consulting regarding user involvement, and the end-users attitudes toward prototyping.

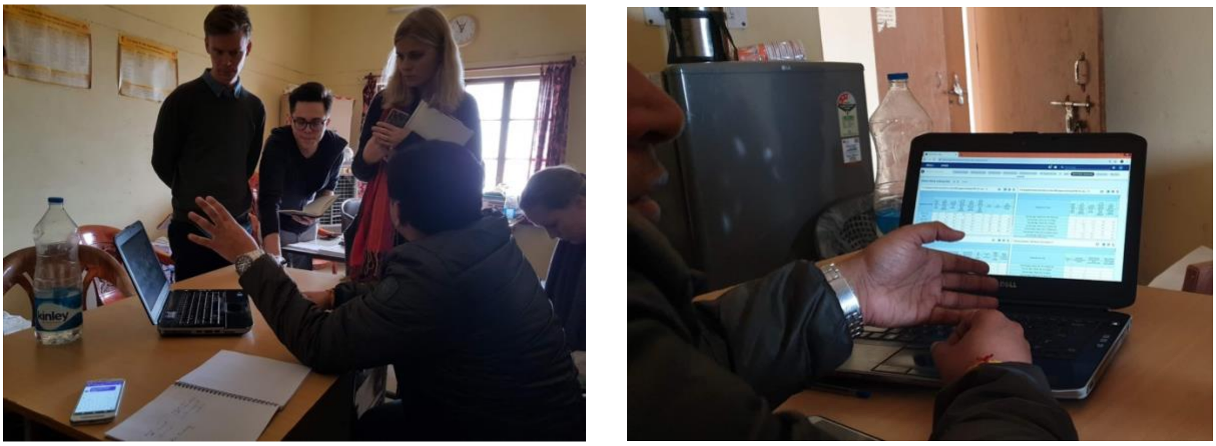

Figure 4.3 Interviews and observations at a health facility

Workshop at the HISP Office

When we were back at the HISP office in Noida, another master's student and I had a workshop in two parts. We gathered a group of employees at HISP India and conducted the workshop over two days, with three-hour sessions both days. This is the workshop I refer to as *workshop 1* in Table 4.6. Both parts of the workshop aimed at introducing a user-centered design focus, and we created personas, prototyped, and discussed various solutions. There were both implementers and developers present, so I also got to observe how they communicated with each other. An overview of the roles of the participants is given in Table 4.7.

Table 4.7 Participants of Workshop 1

Part 1 – 13 participants	Part 2 – 10 participants
4 Implementers 3 Developers 5 Interns 1 UiO master's student	3 Implementers 1 Developer 5 Interns 1 UiO master's student

For me, a goal of this workshop was to lean about the attitudes HISP India's employees have towards prototyping and their reflections regarding conducting similar design activities in their normal practice. The result of workshop 1 is showed in Figure 4.4.

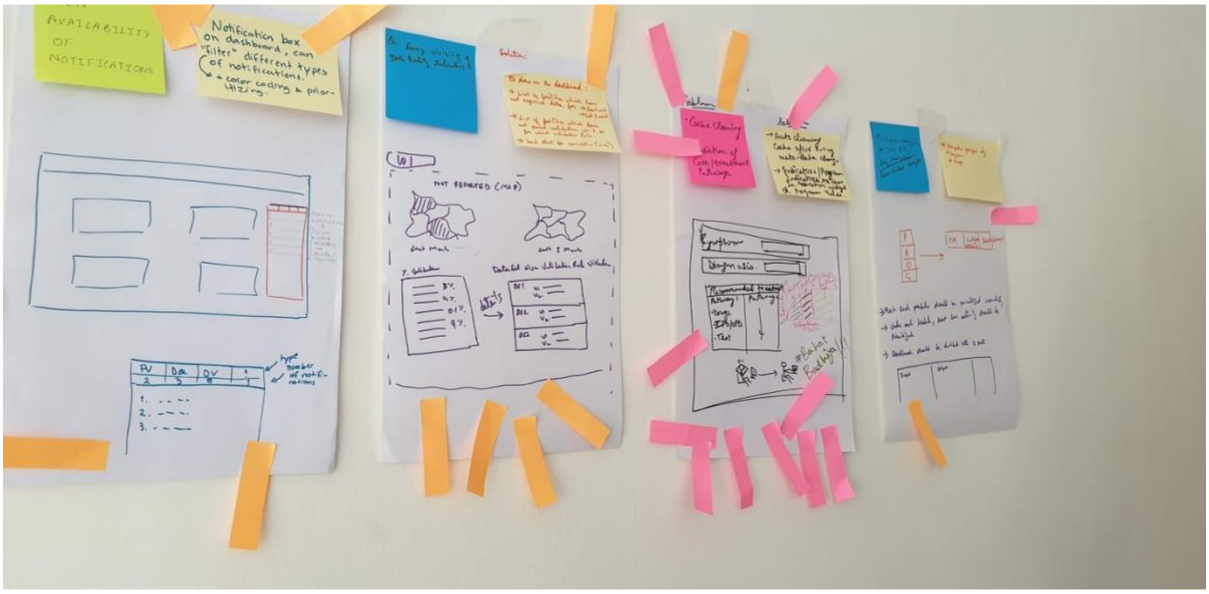
AVAILABILITY OF NOTIFICATIONS
Notification box on dashboard, can "filter" different types of notifications.
+ color coding & prioritizing.
Solution:
Symptoms
Diagnosis

Figure 4.4 Result of workshop 1

Culture and Context

Since we spent a considerable amount of time at HISP India's offices we also experienced the conditions under which they work, such as the frequency with which the power (and therefore also the Wi-Fi) went out. An aspect of the first visit was, in addition to the research, to get an understanding of the Indian culture. I feel like we got on well with many of the employees at HISP India. As a part of getting to know each other we went out to a market and to dinner together after work. We were also taught how to make Indian food by a researcher at HISP India that helped us a lot while we were in India. The aim was, in addition to getting to know the employees, to get a better understanding of the broader context surrounding HISP India's development processes.

4.4.2 Between Field Trips

Between the two trips to India I mainly worked on finding the focus my research and I continued reading relevant literature. My focus then shifted to design processes and tools and techniques in the design processes. I looked at various common techniques that are used when involving users in the design process, such as scenarios and journey maps and considered how likely I thought it was that they fit into the existing practices at HISP India. While in Norway I was part of three workshops: 1) A workshop with the tech-company ABB about design systems, 2) A workshop to test design techniques, and 3) An AMR workshop.

Design Systems Workshop With ABB

This workshop was conducted in collaboration with two designers from ABB with experience in making design systems used in 400 software applications. ABB is a global Fortune 500 company with almost 150 000 employees worldwide. It is therefore important for them that the systems they use look and act the same. It was interesting to get their perspective on design systems as this is something the DHIS2 core team are working on, and both ABB and the DHIS2 core team need to consider many cultures. The designers from ABB told us that their approach was to avoid being "design police" and rather let people use it if they felt wanted to. They also stated that it is important that people's attitudes towards the design system are positive and that they see the value. This workshop gave me important insight into how UCD should be introduced.

Method Workshop

I conducted a pilot workshop with my supervisor and my fellow master's students, in total 6 participants. During this workshop, my goal was to test various design tools and techniques and discuss whether they were applicable to our case in India. I also conducted the workshop to gain more experience with facilitating workshops alone. The techniques we tested included personas, card sorting, and activity models. The personas should be one that we were familiar with from our first field trip, such as a DEO or an M&E. Using the personas each participant wrote down activities that this persona does in relation to their work. This included activities such as "Logs into DHIS2", "has trouble connecting to WIFI", and "complains". The result of the workshop is shown in Figure 4.5.

The main result of this workshop was that we all realized that we did not know as much as we thought about the different user groups we had encountered in India. This helped me reevaluate my research focus and gave me motivation to figure out the challenges with introducing UCD into software organizations. I also realized that I had planned activities for another hour that we did not have time to do because the first activities took up all the time, which made me understand more about how much you can expect your workshop participants to do in a certain amount of time.

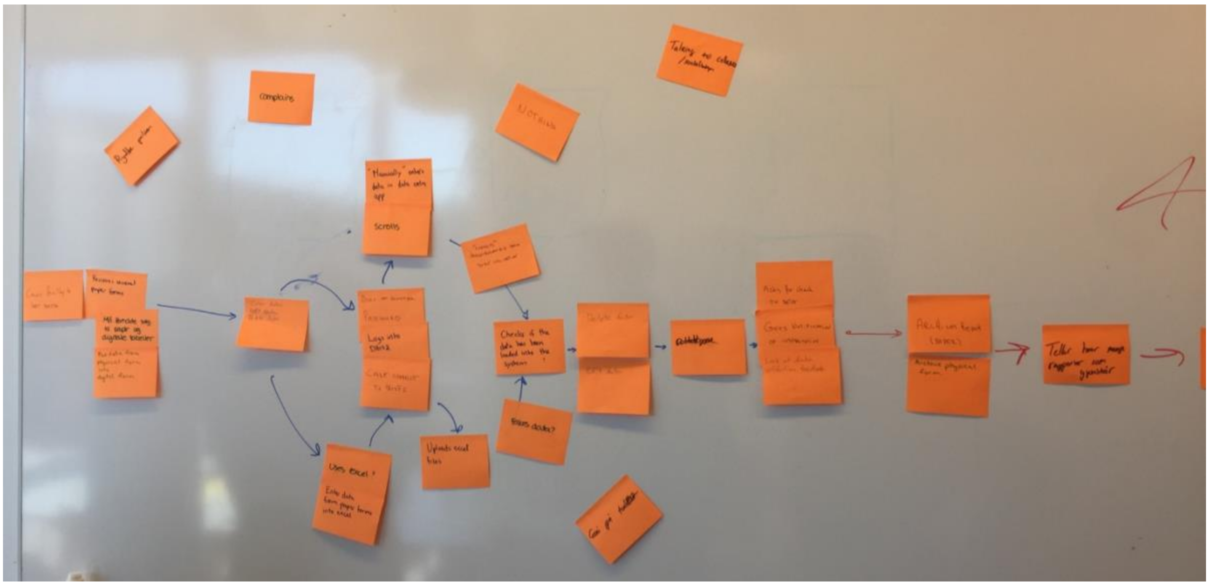
complains
"Manually" enters data in data entry app
Scrolls
Logs into DHIS2
Checks if the data has been loaded into the system?
Uses excel?
Enter data from paper forms into excel
Uploads excel files
Teller hvor mange rapporter som gjenstår

Figure 4.5 Result of the method workshop

AMR Workshop

The AMR project had already started during our first visit to India, but at that time I was not a part of it. The AMR workshop was conducted at the University of Oslo with approximately 25 participants from different domains. HISP India was represented, and so were the client they make the AMR surveillance system for. Other participants included representatives from WHO and other experts on AMR. Attending this workshop gave me more insight into the topic AMR and the state of the AMR project that I hoped to join during my second stay in India.

4.4.3 The Second Fieldtrip

During the first trip to I had not yet decided which problems I could contribute to, and as such was very open to different potential issues. As explained in chapter 4.4.2 my focus shifted between the two field trips, and when I returned to India during the fall of 2019 I was no longer connected to the UP project. While I was not officially included in the AMR project, I tried to establish a presence as a "design consultant". Because of this, I was able to join the two master students who had worked on this project the whole time to visit a health facility in northern India. This trip lasted 7 weeks and I focused more on the existing work practices at HISP India. This trip has three main activists: 1) A field visit to a microbiology lab in northern India, 2) A workshop on the design of locally meaningful technology facilitated by my supervisor, and 3) Interviews conducted with the employees at HISP India in the office. In addition to these activities I observed the everyday situations at the office.

Field Visit to Microbiology Lab

Together with two of my fellow master's students, two newly employed implementers, and an implementer from one of HISP India's remote offices I visited a microbiology lab in the state Himachal Pradesh. This trip happened very spontaneously, and we had no time to think of a strategy in collaboration with the people from HISP India to coordinate our goals and approaches. We interviewed three doctors/professors whose laboratory teams had been using a pilot version of the AMR Surveillance application. This is an application made by one of the master's students as a part of the AMR Project. The purpose of this visit was to get their feedback on the app and how it fit into their practices. Much of the conversation was related to biology and the specific organisms they needed to see in the system, but we also spent a considerable amount of time understanding all the different actors and systems in the ecosystem surrounding the app. This discussion, and the following discussions with the employee's from HISP India, gave me insight into HISP India's focus when collecting requirements. The fieldtrip also gave me more knowledge about the context HISP India's products are used in.

Figure 4.6 Field visit to a microbiology lab

Workshop on the Design of Locally Meaningful Technology

This workshop was the most extensive of all the workshops that are a part of my research. It lasted five days with 8 hours of work every day. This workshop also had participants from HISP India's remote offices in Lucknow and Shimla. In total the workshop had an approximate of 16 participants. As the workshop was quite time consuming not all participants were able to be present all the days. The goal of this workshop was to increase awareness of the importance of end-user involvement when developing and implementing locally meaningful technology. It was also a tool for us as researchers to get an insight into the work practices as HISP India and their familiarity with, and attitudes towards, design processes. The participants were introduced to the concept of usability, various design techniques, and some design processes. They were then divided into groups and given a project to work on for three days. The last two days consisted of presentations and discussions. Table 4.8 gives an overview of the activities each day of the workshop.

Table 4.8 An overview of the activities conducted each day of workshop 6

Day	Activities
Monday	Introduction to the workshop Presentation about the steps of a design processes, how to engage end-users and why you should include end-users Project is presented and groups start group work
Tuesday	Reiteration of the process of brainstorming, prototyping and analyzing findings Groups continue to work on their project
Wednesday	Groups work on their project Groups present their prototypes and design process
Thursday	Group discussions and presentations on the feasibility of the methods and techniques they had been introduced to in a real projects
Friday	Group discussions and presentations on technical implementation methods

My role in this workshop was twofold. All the master's students had been recruited by my supervisor to act as a specific user type with certain activities related to their position. One part of my role was to be an "end -user" and thus be interviewed and involved by the groups in other ways. Another part was to organize two warm-up exercises each day, one at the beginning of the day and another after lunch. Before the workshop I had also prepared a small "booklet" with descriptions of different techniques for user-involvement and understanding requirements (Appendix A). This booklet included descriptions of techniques such as wireframing, guerilla usability testing, and journey mapping.

The project part of the workshop went over three days and I was mainly invited to participate during the first day for "requirements gathering". After this initial meeting with most of the groups I was barely involved further until about 30 minutes before the final presentation the third day to test the prototypes. One of the master's students functioned as the ministry of health and they were kept busy by questions from the groups almost continuously. The "ministry of health" was also responsible for setting up meetings between the "end-users" and the participant groups.

During the third day the groups presented their solutions. Most of the groups had made extensive prototypes in PowerPoint or similar even though we had emphasized that the prototypes should be kept simple. One group even presented a high fidelity, clickable PowerPoint prototype. Other groups, such as the one in Figure 4.7, presented sketches drawn on paper.

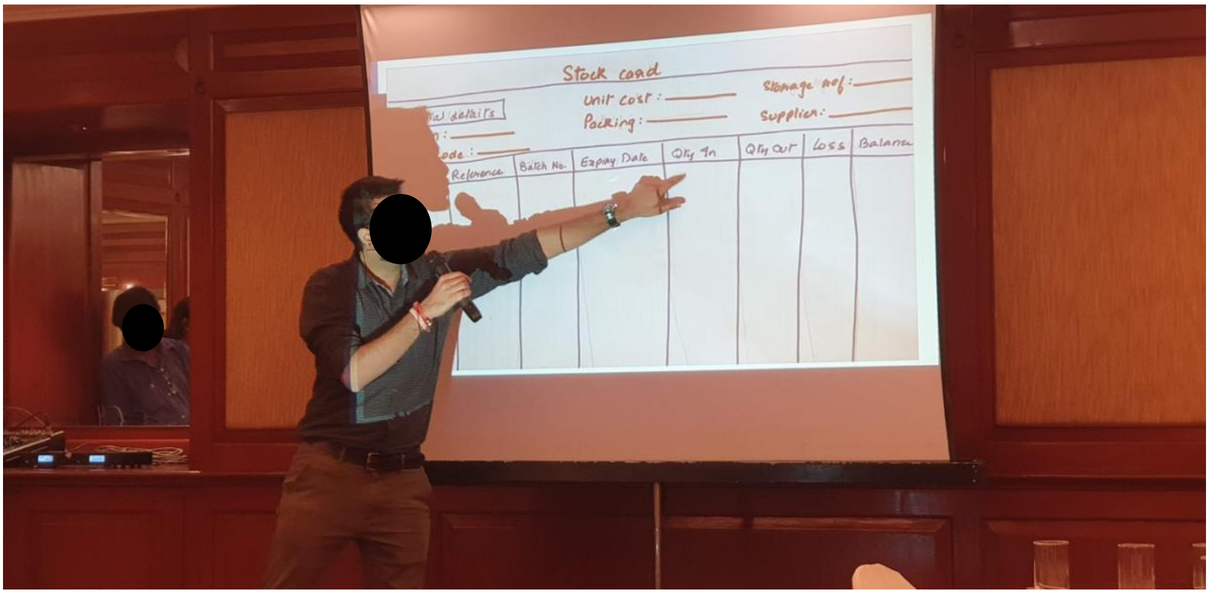
[redacted]
[redacted]

Figure 4.7 Presenting the final prototype

The two final days of the workshop were dedicated to group discussions and presentations. The groups were given a few topics to discuss within themselves, and then presented their findings to the other groups. There were two types of questions, either related to design processes or to the feasibility of the technical implementation. The groups were asked questions such as “Was there any value in engaging with the end-users? Why, and when?”, and “Did you make any assumptions that you should have validated better with the end-users?”. This workshop gave me much insight into the work practices at HISP India, both through observing the project, and by listening to the discussions.

4.5 Methods for Analysis

To analyze data is to make sense of what you observe and present it in a coherent manner that gives value to your observations (Myers, 2013). It can be defined as “a matter of giving meaning to first impressions as well as to final compilations” (Stake, 1995,p. 71, as cited in Yazan, 2015). There is much value to the individual researcher’s impressions, and these are both important to data gathering and in the analyzing process (Stake, 1995, as cited in Yazan, 2015; Walsham, 2006). Stake gives much freedom to the individual researcher to find a suitable way of analyzing data that fit their needs through experience (Stake, 1995, p. 77, as cited in Yazan, 2015). In this section, I will present the modes of analysis I have used to make sense of my findings both during the project and after.

4.5.1 During the Field Work

The process of analyzing data starts as soon as you commence your study, and that data collection and analysis should start at the same time and continue simultaneously (Crang & Cook, 2007). It can be challenging to separate the process of analyzing data into a separate step in your research process as it is intertwined with everything that happens (Myers, 2013). In this sense I have continuously analyzed my findings throughout the project as I have committed to writing field notes and kept a field diary during my visits in India. I have constantly revised my understanding of my findings and the problem area as I have learned more about the context by using the hermeneutic circle as an analytical tool (see Figure 4.8). My focus has been on learning about and introducing the principles of UCD and various design activities into a software organization. But I quickly realized that in order to understand these specific parts of the whole, I had to understand more about the broader context I was in. Getting a better understanding of the DHIS2 ecosystem and general work practices helped me improve my understanding of my research area. My fellow master's students have been very helpful in terms of contextualizing my research area because they have given me an arena to discuss my findings and compare them to theirs which helped me make sense of my data.

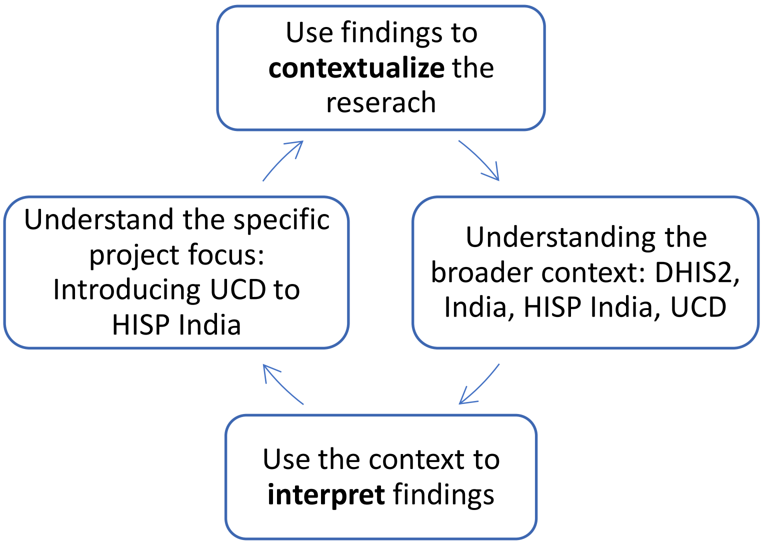
Use findings to **contextualize** the reserach
Understand the specific project focus: Introducing UCD to HISP India
Understanding the broader context: DHIS2, India, HISP India, UCD
Use the context to **interpret** findings

Figure 4.8 Hermeneutic circle for analysis

Sketching at HISP India

During our stay we encountered to a lot of acronyms and various user groups. Another master's student and I therefore started drawing the different actors and the connections between them on a whiteboard in the main meeting room. Our aim was to combine our notes to give us a better understanding of the context. Many of the people who walked into the room and saw our diagram corrected us when they saw that we had made mistakes which helped us get a more correct understanding. The diagram we ended up with is shown in Figure 4.9, and it helped us discuss with the people at HISP India because we then had a shared language.

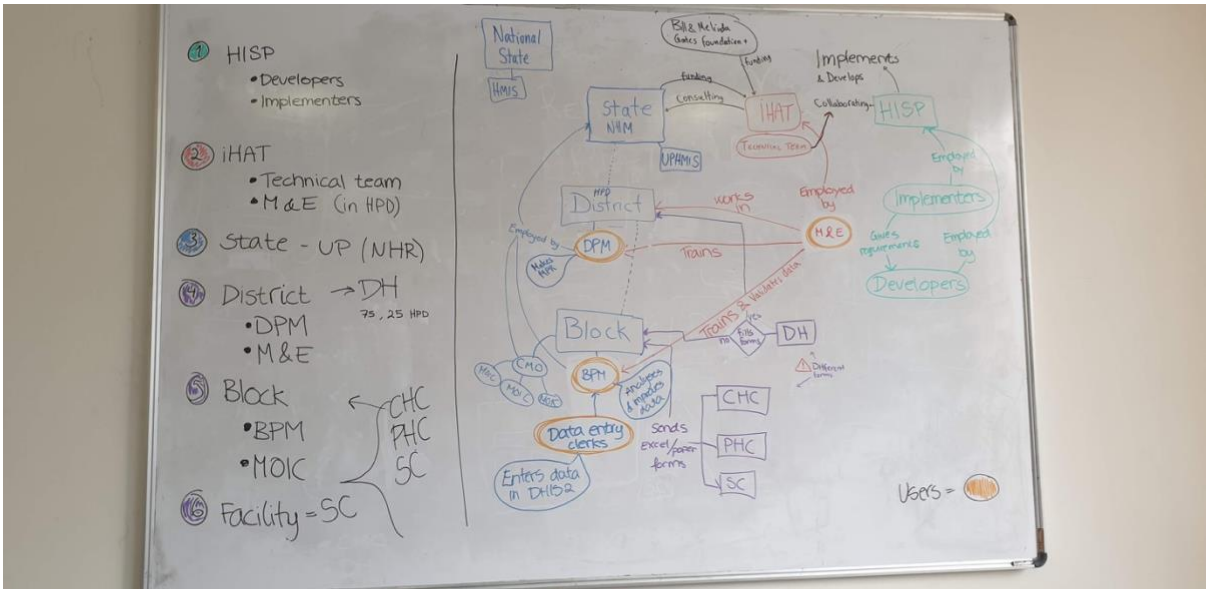

Figure 4.9 Diagram of relationship between actors related to HISP India's projects

4.5.2 After the Field Work

When I returned from the last trip to India, I had learned a lot, but I had not had the chance to properly analyze my findings in such a way that I knew which research contributions I could make. To make sense of my findings I utilized the method thematic analysis. First, I wrote out all my findings and found broad categories like "design knowledge" and "project flow". Then, I used the principles of UCD, and combined them with the categories I had found from my research to get the categories: 1) User involvement, 2) Roles, 3) Design activities, and 4) Process.

4.6 Challenges

During my research project I met a few challenges. Most of these are related to unexpected events and language. This section presents the challenges that have affected my research.

4.6.1 Unexpected Events

A challenge of case studies is that you generally have no control over the situation and the events that unfold (Crang & Cook, 2007), and this has also been the case for me. An example of a situation out of my control happened during my first fieldtrip to India where we were supposed to visit several health facilities but had to cut our visits short because many of the health workers had gone on strike. We thus had to reevaluate our project, and this attributed to the changing of my research focus.

During my second visit we had come to expect that things happened spontaneously and not always on the agreed upon time. What we could not prepare for was that the air pollution in northern India would reach hazardous levels to the point that the air was toxic (see Figure 4.10). This affected our research in two ways: 1) We were affected by the smog and got drowsy, irritated in our eyes and throat, and tired from having to spend weeks on end indoors, and 2) The employees at HISP India worked from home to a greater extent. The combination of these made it hard to make any appointments, such as to conduct interviews, and reduce the amount of "weak" observations we could do at the office.

Figure 4.10 Smog in Delhi

4.6.2 Language and expressions

India has a lot of languages, and although the employees at the office were fluent in english there could be some misunderstandings when we communicated. Expressions that do not translate well and acronyms we were not familiar with are examples of things that could be confusing. For example, I was confused by the term "M&E" as I heard "enemy" and did not understand what this meant. The DHIS2 Design Lab and my fellow master's students were a great asset to help with this confusion as we could compare notes and exchange experiences. The employees at HISP India were also very helpful and took time to explain words and concepts that were confusing.

Different meanings of the words: users, prototype, and design also affected how effectively I was able to communicate. I did not realize that we did not share a collective understanding of these word until late during the second fieldtrip. This might have been avoided if we had a meeting early on to clarify terms that were used by me and by the teams I followed to ensure that we were talking the same language.

During the fieldtrip to the microbiology lab, the three doctors/professors were eager to talk to us and were very comfortable with speaking English. Unfortunately for us, the master's students, the person from the remote HISP office seemed eager to speak Hindi instead and often shifted the conversation over to this. This was a point of frustration for us as we would often have to ask for a translation in a situation where everyone spoke English fluently. It even came to a point where one of the master's students asked a question which one of the professors started to answer in English. The person from the remote office then switched the conversation to Hindi and we never got an answer to the question because the rest of the discussion was carried out in Hindi. When I asked "So, what is the answer to the question?" we got as a reply "Oh, that has been clarified".

We were very fortunate to be invited to come two consecutive days, and because we recognized that we had to be more adamant to keep the conversation in English we came to the office 15 minutes early the second day. This gave us time to small talk and get a bit friendlier with the professor who was present. This helped us establish English as the "official meeting language" and we had much fewer challenges the second day.

Chapter 5 Findings and Analysis

The purpose of this chapter is to present my findings and analyze these in a way that provides insight into the process of introducing UCD in software organizations, especially organizations that specialize in implementing generic software. In this chapter I will first describe the different roles involved in projects at HISP India and the skills these roles generally possess. Then, I present the flow of a general project conducted at HISP India from the project's beginning to its end. After this, I compare HISP India's practices with those of the "ideal" UCD process and present the similarities and the differences between them. Finally, I present the challenges and potential solutions proposed by the HISP India employees related to introducing a UCD approach into their work practices.

5.1 Roles and Related Skills at HISP India

The HISP India team is divided into two main roles: implementers and developers. In this section I will present these two groups and describe the respective skills and areas of responsibility. Common to both groups is that neither have formal education within the field of human-computer interaction (HCI) or similar.

5.1.1 Implementers

Most of the employees that work at HISP India are implementers. The implementers often have domain expertise, such as global health, and are not trained in information technology. When a new project starts the implementers are involved from beginning to end, discussing contracts, negotiating requirements, and doing the appropriate configuration of DHIS2. During the implementation process they are in contact with the client and when a developer is needed, they serve as the link between the client and developer. Most of the implementers are accustomed to leading trainings in the field and are familiar with conducting interviews. Implementers usually work alone on a project.

5.1.2 Developers

There are significantly fewer developers employed than implementers at HISP India. This might be unexpected given that HISP India is a software organization, but it is likely a result of HISP India mainly specializes in configuring the DHIS2 software. In some projects the client requires new functionality that cannot be achieved through configuring DHIS2. This is usually a custom HTML report, but in some cases it also calls for a new application. When this is the case the lead developer reaches out to a developer assigns the project to them. In addition to doing the necessary programming, the developers must also step into the role of the designer. This can be a challenge because none of the developers have formal training or education in interaction design, UCD, or usability. The developers usually work as the only developer on a project. This leads to little transferring of experiences and knowledge within the office. In addition to hindering learning, this way of working can also get very lonely.

5.2 The General Process at HISP India

The general flow of a project from beginning to end usually follows the same four steps. While there are often more intricacies and all projects are different, these four steps are usually the same for all projects. The projects at HISP India can start in one of two ways: 1) A client already knows that they want DHIS2 and contact HISP India directly or, 2) HISP India wins the bidding process. There is work done before they win a specific contract, but I will not present that as I have focused on the events that happen after HISP India have already secured a contract. Figure 5.1 presents the general flow of a project.

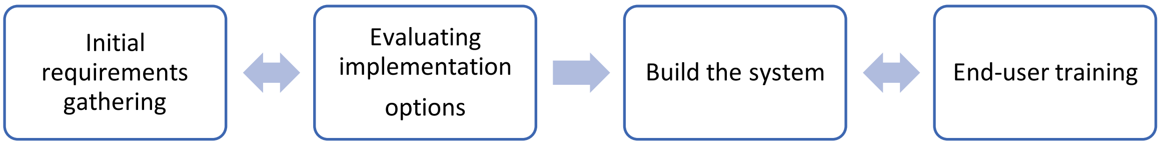
Initial requirements gathering
Evaluating implementation options
Build the system
End-user training

Figure 5.1 Generalized project flow at HISP India

When the contract is secured the first step is to gather the specific requirements from the client that are more in-depth than the general requirements included in the bid. The implementer who is involved in the project contacts the client and propose solutions that satisfy the client's needs. This process is represented by the two first steps in Figure 5.1, and as the double-headed arrow implies, these steps can be repeated several times while the client and HISP India negotiate requirements and solutions. When a solution has been agreed upon, HISP India build the system either through configuring DHIS2 or by making new applications in DHIS2. When the system is finished it is rolled out and end-user training commences. The processes of building the system and end-user training are also connected with a double-headed arrow. This is to signify that HISP India might get feedback while doing the trainings which they can choose to act on and incorporate into the system. The following sections will explain the four steps in more detail.

5.2.1 Initial Requirements Gathering

When HISP India and their client have reached an agreement regarding the contract, the project starts with an implementer going to the client to collect the client's requirements. The requirements of the system are based on two things: 1) What systems does the client already have in place? and, 2) how can

DHIS2 be customized to meet those needs considering the existing systems? The implementer also explores whether the client has any additional requirements.

In general, HISP India's client could be the ministry of health, state level decision makers, or similar authorities. These decision-makers hire HISP India to implement DHIS2 to collect, distribute and analyze data which allows them to increase the precision of their health policies. During this phase the focus is on what output the client needs and how this can be accomplished using DHIS2.

> *«We will gather what exactly is required for their policy-making, for their local decision making, and at what level do they require that.» - Implementer at HISP India*

This step focuses on the formal requirements from the client, and end-users are thus not involved. During this stage it is the client's responsibility to recognize the needs of their users. The end-users of the final product can vary depending on the project. In the case of the UP Project, the state wanted DHIS2 to function as an integrated portal connecting the whole state with the national Health Management Information System (HMIS). This portal, the Uttar Pradesh Health Management Information System (UPHMIS), has end-users on several levels including the state-, district-, and block levels. These different end-users naturally have varying needs. A district-level manager might need dashboards and tables that summarize the data from the entire district, while the data entry operator (DEO) on the block level requires a system that enables easy data entry. Since the client determines to what extent the end-users are involved, the responsibility of conveying the requirements of these different user-groups to HISP India falls on them.

A developer at HISP India explained that while they would like to involve the users in the process it can be challenging to convince their client of the importance of doing so. The employees at HISP India argued that their interest in incorporating design practices is not enough to persuade the client to include the end-users in their projects. For that to happen a developer claimed they would need some "ammunition" that can convince their clients to incorporate user-involvement into the contracts. One developer was concerned that this "ammunition" could then in turn be used against HISP India since it gives the client more knowledge, meaning the clients would expect more from them. There are thus some disagreements regarding how clever it is to use "ammunition" to demand the inclusion of user-involvement in the official requirements.

5.2.2 Evaluating Implementation Options

After the requirements are agreed upon the implementer works through the requirements, planning how they can be fulfilled by DHIS2. The implementer inspects the data elements that are needed to generate

the required output and makes a "prototype" in Excel to keep track of the needed meta data. The required output depends on what kind of a project it is, but usually consists of some kind of report presenting aggregated data, such as the state of various health programs that are implemented across the state. When the implementer is certain of which data elements are needed for the output, they confirm these with the client. If there are any unclarities the implementer reaches out to the client to renegotiate the requirements, hence this is sometimes an iterative process as presented in
Figure 5.2.

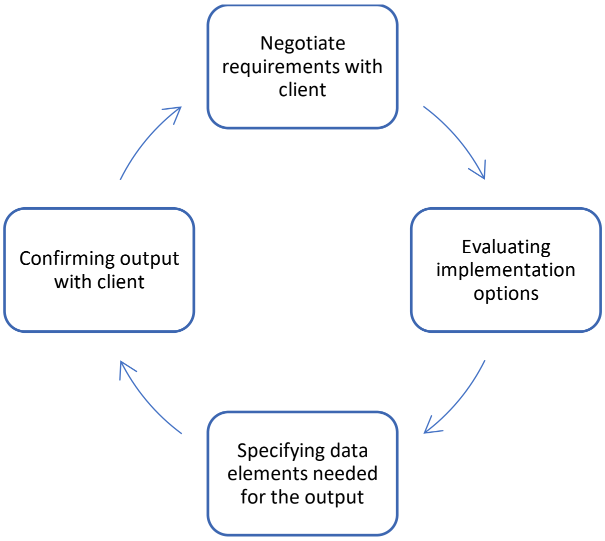
Negotiate requirements with client
Evaluating implementation options
Specifying data elements needed for the output
Confirming output with client

Figure 5.2 Iterative process of specifying requirements

The implementer will usually favor the implementation options that do not require a custom-developed solution specially made for the client. This is to take advantage of the generic aspects of DHIS2. As mentioned earlier, when they are able to leverage existing functionality in the DHIS2 Core, they benefit from not having to maintain said functionality themselves. They also save time as they do not have to build the system from scratch. Due to these considerations the implementer will try to negotiate a solution that can be made using the DHIS2 core functionality.

The end-users of the system are usually not a part of this step as it revolves around the required data elements and is thus output based. Again, the responsibility of representing the users' needs lie with the client. During these iterations the implementer will arrange meeting with the client, and gathers the requirements through interviews, consulting documents, and reviewing current solutions. Prototyping is not used as a tool to communicate with clients. One implementer said that they were scared to show the prototype to the client in case they had misunderstood some requirements, as they generally did not have time to fix any issues. Therefore, they would rather not present the prototypes to the clients for review. Another implementer said that they would not show a low-fidelity prototype to the client, but rather waited until they had a high-fidelity "prototype". This prototype is usually a functioning instance of DHIS2 that has been configured to fit the needs of the client. Prototyping is therefore not seen as a tool to get a shared language and establish a mutual understanding about the requirements, but rather something to impress the client with. Because of this, prototyping is not included as a technique during this stage of evaluation implementation options.

5.2.3 Build the system

When the required data elements are confirmed the implementer starts to configure DHIS2 to fit the client's needs. During this stage there are two general ways to build the system: 1) The implementer can use one of the general packages that DHIS2 natively provides and use predefined options, or 2) They have to customize a solution beyond what the DHIS2 packages provide to meet the requirements of the client. Most projects are of the first category.

> *"Customizing [developing]anything new or designing something new in the system is not the first approach that we take" – Implementer at HISP India*

The most common customization of the software is the development of custom HTML reports, but in more complex cases the developer might have to develop a new application. When this is the case the implementer reaches out to a developer to check for potential solutions to a problem and the lead developer assigns the case to a developer. When a developer is involved, the implementer usually gives them a template of what they need in terms of meta-data and data output and passes the project on to them. The developer then both makes and performs functionality tests on the system themselves. The implementer communicates with the client and serves as the link between the developer and client. The relationship between developer, implementer, client, and end-users can be seen in Figure 5.3.

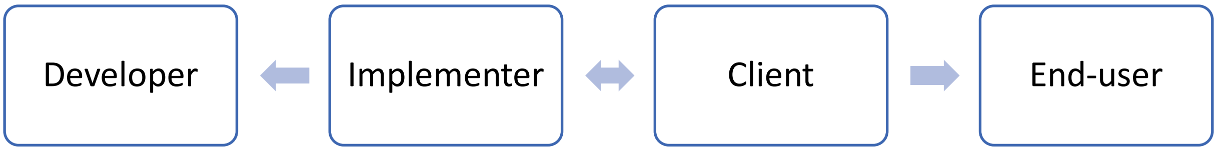

Figure 5.3 Relationship between the actors involved in a project at HISP India

As seen in Figure 5.3, the employees at HISP India do not engage with the end-users during this step as they have to go through the client to get access. It is rare for HISP India to have access to end-users as it is usually not defined in their contract with the client. Their view on the relationship between themselves and their client also affect the level of user involvement, as there seems to be a common misconception that the client is the user. This could explain the attitude I observed from the employees at HISP India that involving users is something the client does as a favor to HISP India and that it is not seen as a valuable tool that is also valuable for the client in the long run.

5.2.4 End-user Training

The final step of the general project flow at HISP India is end-user training which take place when HISP India rolls out the final product. Generally, this is the first time the end-users interact with the system. In some cases, the client asks for a pilot to be tested before the system is introduced to all users. If the pilot satisfies the client, the system is rolled out to the remaining users. During this stage in the process the goal is to teach the end-users how to use the system, not get their inputs on what they would like to include in the system. Usually, no design activities are undertaken to understand the context of use or the users' needs at this stage.

Given the scale of the projects HISP India cannot train all the end-users. Instead, they usually train a few who then train others. The "end-users" trained by HISP India are usually program managers who in turn train the other users of the system. In some cases, these managers are also users of the system, but they do not represent all user groups. In some projects, such as the UP project, there are efforts in place to get feedback from the end-users after the system has been rolled out. In the UP project the health facility workers are part of a WhatsApp group they can use to relay information upward in the system.

> *«It is generally not direct. It is routed through the managers.» - Implementer at HISP India*

This system of WhatsApp groups works like a pyramid, where the health workers are part of one group with their boss. The boss is a part of a different group as well with others on their level and the person who is above them. It is thus the boss' responsibility to forward the feedback from their employees. This is a clear effort to include the end-users at least in some parts of the development process and is a way to get feedback on bugs or other software related issues from a large user group.

5.3 Comparing the HISP India and UCD processes

The previous sections have described the most common stages of a general project completed at HISP India. In this section will analyze my findings compared to the UCD principles *process, roles, user involvement*, and *design activities.*

5.3.1 Process

Like the process at HISP India, the UCD process also has four overall phases: Understand the context of use, gather the user's requirements, make design solutions, and evaluate these solutions against a set of predefined requirements. Based on the evaluations you return to the appropriate step to adjust according to the feedback. This could result in going back one step, or it could mean that you must go all the way back to the first step and get a new understanding of the context of use.

When comparing the project flow of the general project at HISP India and the ideal UCD process we find both similarities and differences. The main similarity seems to be that the processes both consist of requirements gathering and a practice of evaluation. The main difference seems to be that the UCD approach is much more iterative by nature and considers both the context of use and the users of the system to a greater extent. Another difference, as seen in Table 5.1, is that two of the core activities of UCD, *understanding the context of* use and *prototyping* are overlooked by the HISP India team. The UCD process begins with understanding the context of use while the process at HISP India usually starts with negotiating requirements. As presented in Table 5.1, there are some fundamental differences between these two approaches to completing a project.

Table 5.1 Comparing the processes of HISP India and UCD

	HISP India	**UCD**
Project flow	Leaning towards waterfall	Iterative
First step	Gather requirements	Understand the context of use
First solutions	Configured instance of DHIS2	Prototypes
Last step	End-user training	Evaluate design solution against requirements and then building the system

5.3.2 Roles

When comparing the roles involved in a project at HISP India and in an ideal UCD process there are several differences. One discrepancy is that the employees who are responsible for most of the process at HISP India are the implementers who often work alone on a project, not as a part of a multidisciplinary team. The implementers have domain knowledge in subjects such as dentistry or global health, and do not have a technical education. Given that DHIS2 is a health software they could have the relevant knowledge, but it is not necessarily for the specific domain of the new project. No one at HISP India has the training or formal education that qualifies them to act as usability experts, so this role is lacking from their processes.

In an ideal UCD process all the members of the team have met a future user, but this can be very challenging to achieve in practice in HISP India's situation. The role of the end-users in HISP India's process is mainly to provide feedback to the final product. Contrary, in an ideal UCD process the end-users are considered to be a part of the development team and are continuously kept in mind.

Table 5.2 Comparing the roles involved at HISP India with UCD

	HISP India	**UCD**
Teams	Usually only one implementer per team	Multidisciplinary teams
End-users	Give feedback on completed systems	Are considered a part of the development team
Usability expert	Not involved	Should have power to make decisions that affect the usability

5.3.3 User involvement

The level of user involvement is maybe the aspect with the largest gap between HISP India's practice and the ideal UCD process. The ideal UCD process is centered around the potential users of the system, while HISP India's process is centered around the output needed from the system. The main differences are summarized in Table 5.3.

Table 5.3 Comparing user involvement at HISP India with UCD

	HISP India	**UCD**
When is the end-user involved	When they receive training after the system is complete	From the beginning and continuously throughout the project
Techniques used	Interview with client	Interviews, prototypes, personas, scenarios etc.,
Purpose of techniques	Get requirements from client on what output they want from the system	Understanding the use-context and the end-users needs

5.3.4 Design Activities

Because they work contract-based, HISP India are restricted by the agreements they negotiate with their clients. For one of the projects I was involved in, usability testing was one of the tasks the HISP India team were supposed to do according to a requirement specification document. A developer told me that they did not conduct any usability testing and they doubted that anyone else did either, and there was some general confusion surrounding the topic. This exemplifies that it is not enough to include such concepts in the contract if there are not qualified employees to understand the requirement.

Table 5.4 Comparing views of design activities at HISP India with UCD

	HISP India	**UCD**
Prototyping	Rarely done	Should be done continuously
Attitudes toward prototypes	Fun to work with during workshops, but is not used as a tool to communicate with their clients/users	Should be used as a tool to communicate with all involved actors in a project.

5.4 Challenges with Introducing UCD

The question I try to address through my research is how the UCD approach can be introduced to software organizations that specialize in configuring generic software. From interviews, discussions, and workshops I have gathered some potential challenges as seen from HISP India's point of view. First, I will present these challenges as presented by employees at HISP India, and in addition present a few challenges I observed. Then, I go through possibilities identified by HISP India's implementers and developers to introduce UCD into their process and introduce my views on the feasibility of these options. The challenges of introducing a UCD perspective into the projects at HISP India as seen from their perspective is presented in table Table 5.5.

Table 5.5 Challenges of introducing UCD at HISP India introduced by the employees at HISP India

Aspect	**Details**
Attitudes towards end-users	The end-users are reluctant to change and are therefore not willing to participate, and they change every 1-2 years so it is not important to involve them.
Clients as "users"	The client that HISP India have their contract with is regularly referred to as the user. The clients are often busy and do not know the process that goes on in the field where their users work.

Hierarchy in the field	Managers want to be included in the project because they want recognition if the project is successful and do not want people below them to get credit for a project's success. The established hierarchy also limits the people on the lower levels to voice opinions contrary to those of their superiors
Hierarchy in the office	The hierarchy in the office is rather strict. The people in the managerial positions are not always open to suggestions from the people "below" them.
Clients want novel solutions	The clients do not want to pay full price for something if it looks too much like something HISP India has already provided. They do not recognize the work that has been done if it looks too similar to another solution.
Resources	Scarcity of resources such as staff, time, and budgetary restrictions hinder the incorporation of a UCD process.
Generic Software	The generic nature of the software limits the space and options for user involvement.
Available education	There are few courses available at universities that provide a curriculum about UCD/UX/Usability in India.
Willingness to share knowledge	Not all employees are as willing to share knowledge and experiences

5.4.1 Learning UCD

The main obstacle to introducing and working by the principles of UCD at HISP India is the lack of knowledge about these topics amongst the employees. One developer stated that he had looked online to find courses about UX, but had not found anything, and claimed he only found technical courses about programming languages. When asked how they would like to learn, the answers provided by the implementers and developers at HISP India can be divided into two categories: 1) Information resources and, 2) From the experiences of others. As one of the main challenges that hinders UCD being included in their work practices is the lack of knowledge about the topic, I will first discuss the different methods the employees at HISP India stated they would like to learn from. Then, I consider where in the existing process there is room for UCD.

Information Resources

When asked, during a group discussion, what resources would be helpful to learn about UCD and related concepts, the employees stated that books and articles about the concepts would be beneficial. In an individual interview with a developer he said that he doubted anyone would take the time to read any long articles or books. Both developers and implementers stated that an online resource that could present different suggestions for configurations of various use cases would be beneficial. The resource would also include a set of methods and techniques, when and how to apply them, and pros and cons of each method. The final informational resource they mentioned was documentation. Specifically, they wanted documentation that reflected when and why certain decisions were made regarding the design. Table 5.6 gives an overview of the advantages and challenges of the different resources.

Table 5.6 Advantages and challenges of information resources

Resource	Advantage	Challenge
Books and Articles	Gives rich insight and basic introductions to various methodologies, methods and techniques.	Too long and tedious in a hectic workday, and employees have to actively engage with it.
Online Resource	Can provide them with exactly the information they want and need.	Needs to be developed, and employees have to actively engage with it.
Documentation of design decisions	Good documentation is useful both during projects and after. This could give others an opportunity to learn from other people's insight.	This would be a major change in the work practices at HISP India.

The Experiences of others

The learning methods the employees at HISP India seemed most eager about was to learn from others who have had similar experiences. They mentioned a mentor program and the DHIS2 Academies as possible arenas to learn from others. A few implementers addressed the need for more training of new employees as DHIS2 is new to all new employees and takes a little while to learn. One implementer described the situation like this:

> *"Very frankly telling you, there is no formal mechanism of training as such. Nobody is gonna handhold you to learn things. Maybe a one-day session or something if you require, but otherwise you just go over and see that fundamentals course, we go through that and then you struggle for a little time. Catch up!" – Implementer at HISP India*

The implementer then stated that they could see the value of having a mentor program. This could be valuable both in terms of learning DHIS2 configuration and in design-related matters. Another implementer said that as a new employee they would shadow other, more experienced implementers. A formal mentor program could be beneficial to HISP India, and it would have room to teach the new employees about a range of relevant topics, not just DHIS2 configuration, but also UCD and usability principles. This would also increase the knowledge sharing at HISP India, which a developer pointed out as an obstacle to introducing new perspectives and processes. The challenge with a mentor program is that there is no such mechanism already in place at HISP India and would thus have to be established.

Another limitation with this approach to introducing UCD into the organization is that there are no employees who have knowledge about design principles, so this knowledge would have to be sourced from outside the organization. The benefit HISP India has is that the organization is a part of a research collaboration with UiO which gives them the possibility to learn from researchers who visit their offices. Therefore, even though the knowledge must come from outside of the organization it does not necessarily have to be a costly activity, but it does require the employees to have an open mind and be susceptible to learning new skills and adapting the current practices.

As a part of the training a new employee receives, one implementer suggested to send new employees to a "basic level academy" teaching them the basic functions of DHIS2, common problems, and the solutions to common problems as told by other practitioners.

> *"The right approach can be sending a person to an academy which is like a basic level academy or something" – Implementer at HISP India*

The DHIS2 Academies are global arenas for people in the DHIS2 community to meet and exchange experiences. The curriculum of the academies is divided into three levels. First there is the fundamental level. This is a free, online tool that teaches the participants the fundamentals of DHIS2, and most new employees complete this when they start working at HISP India. Then, there are the level 1 courses that cover the key features of the software: Analytics tool, Tracker, and Design & Customization. Although there is an academy about Design and customization, this course focuses more on DHIS2 customization, data sets, elements, and preparation for data analysis, not about techniques for user involvement and usability. Finally, level 2 academies cover more specialized topics, such as Android development, Data quality and Disease surveillance.

The DHIS2 academies could provide a great platform for sharing knowledge and experiences related to UCD. The main challenge with Academies as a source of knowledge about UCD and other design related concepts is that it is governed by HISP on a global level and is thus not something a single implementation organization can initiate on their own. Table 5.7 shows an overview of the advantages and challenges of leaning on the different experience based learning arenas to facilitate learning.

Table 5.7 Advantages and Challenges with the experience based learning arenas

Resource	Advantage	Challenge
Mentor Program	Would facilitate transference of knowledge.	There is no such program today so it would need to be established at HISP India. There are no employees with the required knowledge presently employed at HISP India.
DHIS2 Academy	The academies are already an established arena for knowledge sharing, and is considered to be important by the employees.	The academies are governed by HISP/DHIS2 on a global level and their desire to have an academy about UCD will determine if it can happen

Chapter 6 Discussion

The purpose of this chapter is to discuss the findings described in chapter 5, to answer the following research question:

What are the challenges of introducing UCD into the existing work practices at a software company specializing in the implementation of generic software?

- *What parts of the existing work practices are open to the introduction of UCD?*

My study contributes to a deeper understanding of the challenges of following the UCD methodology as seen by implementers and developers of generic software with limited knowledge about design. There are several challenges that limit the feasibility of successfully introducing UCD into a software organization that specializes in generic health software. Not all the challenges found are specific to implementers of generic software, such as "limited resources" and "lack of knowledge", which are relevant for other types of software organizations as well. This study reconfirms the challenges of user involvement in development processes as found by prior studies (Ardito, 2014; Katsini et al., 2016; Otkjær et al.,2008; Teka et al., 2017) and adds some new insights which might provide a guide for future researchers and professionals who want to further explore the incorporation of UCD principles in practice. A study conducted by Eshet and Bouwman (2016) focused on the challenges of introducing UCD when developing mobile systems. They found that business models often further complicated the introduction of UCD. This thesis confirms that the same challenges are present when investigating the implementation of generic software as well.

In the following sections, I first discuss the implications of the business environment, and the intricacies of generic software. Then, I discuss the effects of the limited design knowledge on the work practices and how UCD can fit into the existing practices. Finally, I address some limitations of this study.

6.1 Complexities of the business and output-based solutions

From the challenges the employees at HISP India mentioned, it becomes apparent that one major challenge HISP India faces regarding introducing UCD is HISP India's business model. One challenging aspect is that most of the projects are contracts for one specific project and the projects are often output-based. Having output-based projects means that the client mainly cares about the data that can be extracted from the system after it is in use. This limits the possibility of following the UCD methodology because the motivation is lacking from the client. This lack of motivation can also be the reason why user involvement is usually not involved in the negotiated contracts. As also found by Teka et al. (2017) in Ethiopia and Ardito et al. (2014) in Italy, the lack of focus on usability in contracts by the government set a standard in the market and is mirrored in the private market. As a result, there is an overall lack of focus in usability and user involvement in IT projects.

Another challenge of contract-based business models in terms of including UCD is that HISP India are engaged with their clients only for a specific project and do not always have a lasting relationship with them. A lasting relationship could be beneficial when trying to convince the clients of the importance of allocating resources for end-user involvement and design activities. This supports the findings of Eshet & Bouwman (2016) who investigated the contract-based projects in a mobile system context. They proposed that user involvement might be easier in situations where the software is more related to a certain user group, such as health. I have found that generic health software faces the same difficulties and obstacles as they reported. The problem seems to be the business model and not that the relevant users are hard to define.

Lastly, given the fact that HISP India works with generic software, most of the solutions are a configuration of generic software and its existing applications. These generic packages are mostly output-oriented, and the required output is decided by project managers depending on the reports they need. This provides complications when user involvement and design activities are to be included in the development process. This compliment the findings of Eshet and Bouwman (2016) who theorized that outcome-based contracts provide a hindrance for usability and user involvement to be a part of contracts, which this study also supports.

6.2 Lack of UCD Awareness and Knowledge

In addition to the complexities that the business model brings, the lack of awareness and knowledge of the importance of UCD is a major obstacle to introducing UCD into the work practices as found in several other studies (e.g. Ardito et al., 2014; Katsini et al., 2016; Otkjær et al.,2008; Teka et al., 2017). Through interviews and discussions, it became apparent that the employees at HISP India do not have any formal or informal education within the fields of UCD, usability, or other relevant concepts. This aspect varies depending on the country the studies are conducted in. Studies conducted in developed countries in Europe do not report this as being a hindrance, while Teka et al. (2017) found this to be a challenge in Ethiopia. Teka et al. (2017) also report "low IT skills of the end-users" is a challenge for user involvement, which is not in accordance with what I found in India. Teka et al. (2017) theorizes low IT skills of the users as a reason for why user involvement only seems to happen after the initial deployment of the software. I found that even when "low IT skills of the end-users" is not a challenge user involvement still only happens at the later stages of the project.

The lack of awareness is a result of limited understanding of the importance of the expertise the end-users can bring to the discussion of requirements, and the value of design activities. The main challenge this causes is that it impedes HISP India's ability to argue for the appropriate resources to be integrated into the contracts they sign. The employees at HISP India were in agreement that for design activities to be put on the agenda in any project it needs to be a specified in the contract, otherwise it cannot be prioritized. My findings support those by Douglas and Liu (2011) that illustrate the lack of general HCI education in the mainstream curriculum in India, and as a result there is a lack of professionals with the required knowledge and a low awareness of the available methods and techniques (Douglas & Liu, 2011). This lack of awareness in turn causes the low investment in UCD in the general business models in India as discussed above.

6.3 How Can UCD Fit into The Existing Practices?

Given that most of the challenges HISP India face are a result of the generic software that they implement and their contract-based projects, it is limited what aspects of the process that are open to the introduction of UCD. Before HISP India can make any adaptions to their process they need to aquire knowledge about UCD, usability, and other related concepts. This can happen through online resources, education, or hiring someone with the required set of skills. My findings also support the findings of Baxter & Sommerville (2011) who stress the importance of introducing UCD and usability with terminology that is familiar to the professionals. It is also important that the approach is developed in collaboration with the professionals who will adapt a user centered focus. Finally, it is important to consider the overall value generated by introducing UCD and other design activities compared to the time and resources invested.

Based on my experiences, and given the relationship HISP India has with UiO, one opportunity for gaining insight into the UCD process is to include students from UiO who have the appropriate education into a project as an equal member of the team from start to finish. This approach requires a lot of preparation and facilitating in advance but has potential to introduce UCD in a less costly and more organic way.

Another option is to create online resources that introduce the implementers and developers to tools and methods that are relevant to their practices and needs. For example, an implementer requires knowledge about methods that improve their understanding of the context of use and the end-users, such as personas and scenarios. Although these methods could be useful to the developers as well, the most relevant methods for the developers could be methods directed at increasing the usability of the user interface, such as heuristic evaluation or usability testing. The employees expressed that such a resource could be useful if it was adapted to fit their process. For example, the resource should provide the employees with facts they can use to encourage their clients to include user involvement and design activities in the contract. Most of the employees agreed that for such a resource to be used it must be quick to use to be accepted into their practice.

It also seems that for UCD to be put on the agenda, the culture at HISP India needs to change to be more open. Among the employees I spoke to, those that were the most engaged and interested in learning about design activities have since quit or reduced their employment percentage. One developer said that they felt that they were judged for wanting to prototype before developing, and that they felt the others saw it as unnecessary to make sketches. This developer said that he was the only one that he had seen at the office who actively used the whiteboards to sketch anything before he developed it. This developer resigned shortly after our first visit. There seems to be an interest in adjusting the development process, but factors such as time and resources restrict the perceived possibility of this being successful. This could imply that the management of HISP India should get more involved and encourage their employees to perform design activities and learn more about relevant design concepts.

When there are employees at HISP India with the required knowledge, the two aspects of the general flow of a project at HISP India that are the most open to the introduction of UCD are: 1) The period of evaluation of implementation options, and 2) When developers make new applications and HTML reports. Both activities happen within the HISP India office, and do not rely on the client, but rather depend on the employees' motivation to incorporate aspects of UCD into their process. Ideally, the end-users should be involved, but given the business model it is more challenging to alter activities that involve the client. In time, if the employees at HISP India gain more knowledge about UCD they should also aim to include design activities in the contracts they negotiate with their clients.

6.4 Limitations

This study has not gone in depth in the research of the culture in which the software was used, which could be a relevant factor. Research suggests that methods used in usability testing and design methodologies that are made in a western context do not fit into other contexts. This study has not considered the cultural aspects to a great extent. I have only focused on one software organization in India and gone in depth trying to understand the specific organization's practices. This might affect the possibility of generalization of the findings. Given the qualitative approach to data collection and analysis it will also be difficult to replicate this study.

Chapter 7 Conclusion

The UCD methodology is valuable because it has the potential to reduce the cost of development, and to increase the users' satisfaction with new products. In a HMIS context, following a UCD process can result in increased data quality and therefore helps decision-makers provide better health care programs. Many software systems developed for the health context are generic systems. This poses new challenges when following a UCD process, both for the vendors of the generic software, and for the organization that adapts the software to fit their local context. In the empirical case of this thesis, I have followed an organization located in India who customize the generic health software DHIS2 on behalf of their clients. Based on the findings from a case study of the work practices at this software organization in India, this thesis has highlighted challenges faced when trying to introduce UCD into existing work practices, as well as where in these practices there is room to introduce UCD.

The results of this thesis can help other researchers with their studies. The results are especially relevant to other members of the DHIS2 Design lab, who can benefit from the identified challenges related to UCD in software organizations, both from existing literature and new additions from my fieldwork, in the context of HISP India. Theoretically, this thesis contributes by adding to the list of known challenges associated with introducing UCD in software organization. Further, I suggest that the learnings from studies done on the business complexities surrounding the development of mobile systems (Eshet & Bouwman, 2016) is transferrable to the context of generic health software as the two contexts have similar obstacles. Lastly, this thesis has also identified which aspects of the general project flow at HISP India that might be open to the introduction of UCD.

7.1 Further Research

Based on the findings of this thesis, several interesting topics for further research emerge. Firstly, this thesis has investigated the practices of only one software organization that specializes in implementing generic software. It would be interesting to investigate similar organizations, both in India and in other countries to further explore how UCD can be introduced into these kinds of organizations.

Further, the design of online resources aimed at teaching employees at various HISP groups about design techniques and methodologies is an intriguing and important avenue of investigation. This is currently being undertaken by another master's student at UiO.

References

About Us. (n.d.-a). *Health Information Systems Program*. Retrieved June 22, 2020, from http://www.hisp.org/index.php/about-us/

About Us. (n.d.-b). Retrieved June 21, 2020, from https://hispindia.org/Pages/About_Us/About_Us.html

Antimicrobial resistance. (n.d.). Retrieved June 21, 2020, from https://www.who.int/health-topics/antimicrobial-resistance

Ardito, C., Buono, P., Caivano, D., Costabile, M. F., & Lanzilotti, R. (2014). Investigating and promoting UX practice in industry: An experimental study. *International Journal of Human-Computer Studies*, *72*(6), 542–551. https://doi.org/10.1016/j.ijhcs.2013.10.004

Ardito, C., Buono, P., Caivano, D., Costabile, M. F., Lanzilotti, R., & Dittrich, Y. (2014). Human-Centered Design in Industry: Lessons from the Trenches. *Computer*, *47*(12), 86–89. https://doi.org/10.1109/MC.2014.355

Baxter, G., & Sommerville, I. (2011). Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, *23*(1), 4–17. https://doi.org/10.1016/j.intcom.2010.07.003

Braa, J., Monteiro, E., & Sahay, S. (2004). Networks of Action: Sustainable Health Information Systems across Developing Countries. *MIS Quarterly*, *28*(3), 337–362. JSTOR. https://doi.org/10.2307/25148643

Buur, J., & Bødker, S. (2000). From usability lab to “design collaboratorium”: Reframing usability practice. *Proceedings of the Conference on Designing Interactive Systems Processes, Practices, Methods, and Techniques - DIS ’00*, 297–307. https://doi.org/10.1145/347642.347768

Collect, Manage, Visualize and Explore your Data | DHIS2. (n.d.). Retrieved June 21, 2020, from https://www.dhis2.org/

Crang, M., & Cook, I. (2007). *Doing Ethnographies*. SAGE.

Damodaran, L. (1996). User involvement in the systems design process-a practical guide for users. *Behaviour & Information Technology*, *15*(6), 363–377. https://doi.org/10.1080/014492996120049

DHIS2 design lab. (n.d.). Retrieved June 22, 2020, from https://www.mn.uio.no/ifi/english/research/networks/hisp/dhis2-design-lab/index.html

Douglas, I., & Liu, Z. (Eds.). (2011). *Global usability*. Springer.

Eshet, E., & Bouwman, H. (2016). Context of Use: The Final Frontier in the Practice of User-Centered Design? *Interacting with Computers*, iwc;iww030v1. https://doi.org/10.1093/iwc/iww030

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., & Cajander, Å. (2003). Key principles for user-centred systems design. *Behaviour & Information Technology*, *22*(6), 397–409. https://doi.org/10.1080/01449290310001624329

Karat, J. (1997). Evolving the scope of user-centered design. *Communications of the ACM*, *40*(7), 33–38. https://doi.org/10.1145/256175.256181

Katsini, C., Avouris, N., & Lanzilotti, R. (2016). Usability Engineering Practices in Software Development Organizations: The Greek and the Italian Case Study. *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI ’16*, 322–323. https://doi.org/10.1145/2909132.2926085

Klein, H. K., & Myers, M. D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly*, *23*(1), 67–93. JSTOR. https://doi.org/10.2307/249410

Kujala, S. (2003). User involvement: A review of the benefits and challenges. *Behaviour & Information Technology*, *22*(1), 1–16. https://doi.org/10.1080/01449290301782

Li, M. (2019). *An Approach to Addressing the Usability and Local Relevance of Generic Enterprise Software*. 16.

Merriam, S. B. (1998). *Qualitative research and case study applications in education.* San Francisco, CA: Jossey-Bass

Myers, M. D. (2013). *Qualitative Research in Business and Management*. SAGE Publications. https://books.google.no/books?id=fEIaDjIiF7IC

Norman, D. A., & Draper, S. W. (1986). *User Centered System Design: New Perspectives on Human-computer Interaction*. Taylor & Francis. https://books.google.no/books?id=Qz5jQgAACAAJ

Noyes, J. M., Starr, A. F., & Frankish, C. R. (1996). User involvement in the early stages of the development of an aircraft warning system. *Behaviour & Information Technology*, *15*(2), 67–75. https://doi.org/10.1080/014492996120274

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research*, *2*(1), 1–28. JSTOR.

Otkj, J., Nguyen, K., Risgaard, P., & Stage, J. (2008). *Obstacles to Usability Evaluation in Practice: A Survey of Software Development Organizations*. 10.

Rosalind Edwards, & Janet Holland. (2013). *What is Qualitative Interviewing?* Bloomsbury Academic.

Simonsen, J., & Robertson, T. (2012). *Routledge international handbook of participatory design*. Routledge; /z-wcorg/.

Stake, R. E. (1995). *The art of case study research.* Thousand Oaks, CA: SAGE Publications.

Teka, D., Dittrich, Y., Kifle, M., Ardito, C., & Lanzilotti, R. (2017). User Involvement and Usability Evaluation in Ethiopian Software Organizations. *The Electronic Journal of Information Systems in Developing Countries*, *83*(1), 1–19. https://doi.org/10.1002/j.1681-4835.2017.tb00616.x

Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems*, *15*(3), 320–330. https://doi.org/10.1057/palgrave.ejis.3000589

What is User Centered Design? (n.d.). The Interaction Design Foundation. Retrieved June 22, 2020, from https://www.interaction-design.org/literature/topics/user-centered-design

Yazan, B. (2015). *Three Approaches to Case Study Methods in Education: Yin, Merriam, and Stake*. 20.

Yin, R. K. (2002). *Case study research: Design and methods. Thousand Oaks,* CA: SAGE Publications

Appendix A

Design Techniques

& How to use them

Semi Structured Interview

The interview is a way to collect qualitative data and understand why users behave as they do. You can interview people in groups or one-to-one, depending on what your purpose with the interview is.

Steps

1. Recruit interview persons.
2. Plan the interview and create a script or a list of topics you want to cover. Start with open questions to warm up the interviewed person, continue with harder questions that require more thought (if you have such) and return to open questions at the end so the interviewed person feels comfortable.
3. Conduct a pilot interview to test your script and rehearse your interviewing skills. You can also get someone to review your scripts before the interviews.
4. Conduct the interviews. If you want to record, ask for permission. Get another person to take notes during the interview. Ask the interviewed person “why is that” or “how do you mean” to get them to provide details. The interviewer should take an apprentice role, ask about things that occur and take notes.
5. Analyze and report your findings. Transcribe recordings and analyze your notes.

Participatory Observation

A method that is relatively cheap to carry out and needs few resources. However, it can often be very time consuming and longitudinal. There are many types of observations, this description is of participatory observation.

Steps

1. Sit with the participants and observe while they perform certain tasks. Do not interfere with the tasks you have asked the participants to perform.
2. Watch carefully while taking notes.
3. Ask the participants to elaborate on what they are doing and ask follow-up questions. One way to do this is the “think-out-loud” approach. Ask the participant to do their tasks while saying everything they think out loud. The observer should take an apprentice role, ask about things that occur and take notes.
4. Take many notes. Write about everything - you never know what might prove valuable during data analysis. Include both what the participant says **and** what they do. It’s important to understand what the participant believes to be true, *even when it’s not technically correct.*
5. When the observation is finished, write a summary that highlights the key insights. If you have observed multiple people, write an individual summary before writing an overall summary.

Brainwriting

A method for coming up with a lot of ideas and solutions to problems. When brainwriting participants write down ideas individually before discussing them. Research says this makes groups come up with more and better ideas than through traditional brainstorming.

Steps

1. Before starting this exercise you should have defined some problems.
2. Start by handing out post-its and a pen to each of the participants (both researcher and end-users).
3. Spend ten minutes, individually, writing or drawing as many ideas as possible. *Don't think just write.*
4. After the ten minutes are up, each participant takes their post-its and hangs them up on the wall. Make sure there is space between each post-it.
5. Take some time to read through all of the post-its individually. For each of the ideas on the wall, each participant must write a new idea to build upon the existing one.
6. Repeat until you are satisfied with the number of ideas. As a group (both researcher and end-users), cluster the ideas and prioritize the best ones.

Wireframing

A wireframe is a two-dimensional illustration of a page’s interface that specifically focuses on space allocation and prioritization of content functionalities available, and intended behaviors. For these reasons, wireframes typically do **not** include any styling, color, or graphics. This technique does not require involvement of users, but keep in mind who the system will be used by.

Steps

1. Build preliminary blueprints that show structure, placement, and hierarchy for your product. Steer clear of font choices, color, or other elements that would distract both the researcher and the reviewer. Lightweight designs are conceptually easier på reconfigure.
2. Use this opportunity to start listing what UX/UI patterns you will need. For example, if you use a design system, what atoms or molecules can be used.
3. Review your wireframes with specific user scenarios and personas in mind. Can users accomplish their tasks with the wireframe you are sketching out?
4. Use the wireframes to get the team’s feedback on feasibility and structure.

Guerilla Usability Testing

Guerrilla usability testing is a way to evaluate how effective an interface is by testing out its visual design, functionality and general message on its intended audience and capturing their responses. A common mistake in user research is performing usability studies too late in the design process. Early testing helps teams validate their ideas **before** they spend time and resources on building them out, and has proven to be cost-effective in the long run.

Steps

1. Define what aspects of your product you want to test.
2. Prepare what you need to conduct the test (wireframes, sketches etc.)
3. Once you have come up with your tasks, turn them into scenarios to pull them together and make them more accessible for participants to follow. For example a scenario could be *“Buy a pair of green socks”.*
4. Find relevant participants and observe as they do the tasks you have prepared.
5. Analyze your findings and make the relevant changes to your sketches.

Personas

Personas are fictional characters created based upon the user research to represent the different user types that will use the service/ product/ site. Creating personas helps to understand users, their needs, experiences, behaviors, and goals.

It will give you a better understanding of who the end-users are. A persona can be represented by a form such as the one on the right. You can add or remove features to make it fit into your project.

Photo

Personal motto

Goals and motivations

Name

Age

Occupation

Bio/description

Challenges and frustrations

Personas

Personas are fictional characters created based upon the user research to represent the different user types that will use the service/ product/ site. Creating personas helps to understand users, their needs, experiences, behaviors, and goals.

It will give you a better understanding of who the end-users are. A persona can be represented by a form such as the one on the right. You can add or remove features to make it fit into your project.

Photo

Personal motto

Goals and motivations

Name

Age

Occupation

Bio/description

Challenges and frustrations

Journey Map

A method for visualizing the steps taken by users as they engage with a product or service. A customer journey focuses on what kind of experience the users have at different stages and what activities they do. Mapping a customer journey will help you to understand your users’ pain points, delight points, and needs.

Steps

1. Set the scope for the mapping - what parts of the service do you want to focus on? Select a target group. Decide on who to include in this exercise; it could be stakeholders who know their users - or even better - actual users. If you already have some insights, it is a good idea to prepare a mock-up of the journey and use it as a starting point in the workshop.
2. Identify all the steps the users can encounter and illustrate them. Examples of these can be actions, touchpoints, feelings, emotional curves, needs, etc.
3. Have the participants fill out the journey with post-its. Make sure to base it on actual user insights and needs. Include discussions so that everyone agrees on the end result.