Design Labs in Enterprise Software Ecosystems: An Approach to Strengthen Software Usability

Anders Erik Brustad and Hanna Alderslyst Kongshem

Thesis submitted for the degree of
Master of Informatics: Design, Use, Interaction
60 credits

Department of Informatics
The Faculty of Mathematics and Natural Sciences
University of Oslo

Spring 2021

Design Labs in Enterprise Software Ecosystems: An Approach to Strengthen Software Usability

Anders Erik Brustad and Hanna Alderslyst Kongshem

2021

2021

Design Labs in Enterprise Software Ecosystems: An Approach to Strengthen Software Usability

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

Printed: Reprosentralen, University of Oslo

Abstract

Increasingly, software implemented in organizations are generic enterprise solutions, designed to fit general use rather than specific users. The fact that enterprise software is made for general use, makes the established practice of designing for usability incompatible for designing generic software. As a result, making enterprise software that is perceived as usable is recognized as a challenge. One approach to address usability-related challenges discussed in academic literature are design labs that emphasize user involvement, usability testing and collaborative efforts. However, existing conceptualizations of design labs are ill-equipped to address usability-related challenges in generic software due to the scale and diversity in user contexts. Using design labs as a means to address usability within the context of enterprise software ecosystems is an unexplored topic, and thus represents a gap in the literature.

This thesis examines what roles a design lab can play to strengthen the software usability within enterprise software ecosystems. By exploring the challenges vendor and implementation partners face when addressing usability, we identify potential ways a design lab can remedy these challenges. Through a one-and-a-half-year embedded case study we followed the DHIS2 Design Lab, which attempts to address usability-related problems through strengthening both the development of the generic software, and the processes of implementing the software in local use contexts.

Based on our empirical case, we contribute to literature on enterprise software ecosystems and design labs by conceptualizing a generic software design lab, which takes into account the scale and diverse contexts of use of generic software. We further contribute by identifying four roles a design lab can play to address usability-related problems in generic software ecosystems. In addition to being relevant to researchers, our conceptualizations and findings are relevant to practitioners concerned with design in enterprise software ecosystems.

Keywords: design lab, generic-level design, implementation-level design, design infrastructure, generic software, enterprise software ecosystems, usability

Acknowledgments

First and foremost, we would like to express our deepest appreciation to our supervisor, Magnus Li, whose knowledge and guidance has enabled us to sharpen our minds and stay motivated throughout the writing of this thesis. We could not have done this without your persistent help and encouragement.

We would like to show our gratitude to all the participants of our research project. Especially, we would like to thank HISP Malawi and Saudigitus for their warm welcome, and for taking the time to talk with us. This has made considerable contributions to our research.

The DHIS2 Design Lab and all its members has been an invaluable resource for us throughout this process. Thank you for all the engaging conversations, feedback and drawn-out lunches. We would especially like to thank Bendik for being a great travel partner and source of encouragement, jokes, and laughter throughout the last couple years. You are truly a great friend. We would also like to express our gratitude to our fellow design students on the seventh floor at IFI.

We would especially like to thank friends and family for their sympathetic ear and unconditional support throughout the process. Anders would especially like to thank Alma for all the support, encouragement, and patience.

Hanna and Anders
University of Oslo
June 2021

Table of Contents

List of Figures

List of Tables

Abbreviations

CbC	Construction by configuration
DHIS2	District Health Information Software 2
ERP	Enterprise Resource Planning Systems
HIS	Health Information System
HISP	Health Information Systems Program
HMIS	Health Management Information System
IS	Information System
ISG	Implementation specialist group
MoH	Ministry of Health
NGO	Non-Governmental Organization
PD	Participatory Design
UCD	User-Centered Design
UI	User Interface
UiO	University of Oslo

Chapter 1 - Introduction

This thesis examines what roles a design lab can play to strengthen the software usability within an enterprise software ecosystem. Based on empirical data collected through a one-and-a-half-year embedded case study we have explored the challenges an enterprise software vendor and implementation partners face when addressing usability-related issues, and potential ways a design lab can remedy these challenges. More specifically, we conceptualize a generic software design lab and identify four roles a design lab can play to strengthen the software usability in an enterprise software ecosystem. In the empirical case of this thesis, we follow the DHIS2 Design Lab which is an initiative established at the University of Oslo. The DHIS2 Design Lab aims to explore how design and innovation can be facilitated and promoted in the implementation processes of the health information management system, District Health Information System 2 (DHIS2) (UiO, n.d.-a).

1.1 Motivation

An increasing amount of software implemented in organizations today is generic enterprise solutions, often referred to as "off-the-shelf software" or "packaged software" (Bansler & Havn, 1994; Sommerville, 2008; Strong & Volkoff, 2010). Generic software is designed to fit general use rather than a specific user, and has the benefit of rapid implementation and low costs (Bansler & Havn, 1994). The development process of generic software can be described as a two-step approach with a vendor developing the generic software, and the customer, or implementation partner being in charge of implementation and configuration (Bansler & Havn, 1994). This process is part of what Dittrich (2014) conceptualizes as an *enterprise software ecosystem*. The term enterprise software ecosystem describes the processes of software development, implementation and configuration, the actors involved, as well as the relationship between these. In this thesis, *enterprise software ecosystems* refer to the environment where these processes and relations take place, and *generic software* refers to a specific software with generic attributes.

Design of generic software is however not exempt from difficulties. A comprehensive body of research emphasizes the importance of use-oriented design, considering end-users needs and practices (Ehn, 2008; Kujala, 2003; Li, 2019b; Norman, 2013; Rosson & Carroll, 2007) Having this focus is argued to be of high relevance in terms of software usability, enabling the users to achieve a desired efficiency, effectiveness and user satisfaction (Grudin, 1991; ISO, 2018). The established practice of designing for usability is however incompatible when designing generic software, which aims to fit general use, rather than specific users. As a result, making generic software that is perceived as usable is recognized as a challenge (Li, 2021; Martin et al., 2007; Sia & Soh, 2007; Strong & Volkoff, 2010).

Design labs is a well-established concept in HCI and IS literature, and aims to describe an approach to address usability related problems, and several examples show that such labs are successful in accomplishing this goal (Andersen et al., 2018). Through user involvement, usability testing and collaborative efforts between designers and users situated in a design lab, technology projects can result in usable software products. There are several types of design labs conceptualized and covered by design lab literature to this date, including the usability laboratory (Nielsen, 1994), the design collaboratorium (Buur & Bødker, 2000), the Design:Lab (Binder & Brandt, 2008), and living labs (Alavi et al., 2020). While they all emphasize the importance of usability, they have inherent characteristics separating one from another. For instance, the literature documents examples of living labs used to direct a focus towards exploring elements of a design related to the context of use (Andersen et al., 2018). Usability laboratories are conceptualized as an effort to address usability specifically through usability testing (Nielsen, 1994), and through design experiments and collaborative efforts (Binder & Brandt, 2008; Bødker et al., 1995; Buur & Bødker, 2000).

However, the existing body of literature is mainly concerned with the use of design labs in small-scale projects. In these projects, designers and developers can work closely with end-users to create custom solutions, fit to their needs. Design of software within enterprise software ecosystems represent a very different, yet relevant context. However,

using design labs as a means to address usability related problems within the context of enterprise software ecosystems is an unexplored topic, and thus represent a gap in the literature (Li, 2019b). We address this gap by outlining four roles a design lab can play to strengthen usability in an enterprise software ecosystem.

1.2 Research Question

Based on the described problem of making generic software that is perceived as usable, the following research question is defined for this thesis:

What roles can a design lab play to strengthen the software usability within enterprise software ecosystems?

Roles refer to different functions a design lab can play. Usability refers to how usable a software is for the users in terms of user satisfaction.

Through an embedded case study, we have investigated the DHIS2 Design Lab and the enterprise software ecosystem it is a part of, further referred to as the *DHIS2 ecosystem*. To address the research question, an exploration of two central stakeholders in the DHIS2 ecosystem was essential: the *DHIS2 core team* who are in charge of the development of the generic core software of DHIS2, and the *implementation specialist groups* (ISGs) who are the implementation partners in charge of the implementation and configuration of DHIS2 to fit specific organizational contexts. Challenges they experience in their work to address usability provides a foundation for our understanding of how the DHIS2 Design Lab can remedy these challenges, thus resulting in an understanding of what roles a design lab can play to strengthen the software usability in its respective enterprise software ecosystem. To address this thesis research question, we have outlined and followed four main objectives:

1. Explore the challenges the DHIS2 core team and ISGs face when addressing usability

2. Explore the current practices of the DHIS2 Design Lab and how it works to address usability
3. Explore the potential ways the DHIS2 Design Lab can remedy the usability challenges that the DHIS2 core team and ISGs experience
4. Identify and discuss roles a design lab can play to strengthen the software usability in enterprise software ecosystems based on our empirical case

Based on our empirical findings and existing enterprise software ecosystems and design lab literature, we conceptualize a generic software design lab, and identify and discuss four concrete roles a design lab can play to strengthen software usability in the context of enterprise software ecosystems. From this study we make both practical and theoretical contributions. We make a theoretical contribution by outlining the four roles, which we view as an extension to the already established lab-type approaches. Additionally, the roles make a practical contribution by serving as a starting point for new generic software design labs, or serving as a fulcrum for taking existing labs in a direction closer to enterprise software ecosystems. Additionally, we contribute to the DHIS2 ecosystem by providing a concrete overview of new possibilities the DHIS2 Design Lab can take in the future. Moreover, our empirical insights support the previously reported challenges in addressing usability in enterprise software ecosystems in IS research, and provides insight into one approach for addressing these issues.

1.3 Chapter Summary

This thesis is structured in the following chapters:

Chapter 2 - Background

Provides a general background and essential information related to this research project. The Health Information Systems Program (HISP), the District Health Information System (DHIS2) and the DHIS2 Design Lab is introduced and described in this chapter.

Chapter 3 - Related Research

Two streams of relevant research are introduced to provide an understanding of: 1) different Design Labs conceptualized by literature as methods to address usability issues,

2) working with software usability in enterprise software ecosystems. Lastly, the theoretical lens of this research project is outlined.

Chapter 4 - Research Approach

The ontological and epistemological assumptions, methodology and data collection methods for this study are described and justified. We also outline the data analysis process.

Chapter 5 - Findings

Provides our empirical grouped in three main sections: challenges for addressing usability, the DHIS2 Design Lab and its position in the DHIS2 ecosystem, and opportunities for strengthening the software usability.

Chapter 6 - Analysis and Discussion

Presents the contributions of this thesis, which consists of a conceptualization of a design lab relevant within generic software, and four roles a design lab can play to address usability-related problems in enterprise software ecosystems.

Chapter 7 - Conclusion

Concludes the thesis by summarizing the findings and contributions.

Chapter 2 - Background

This chapter is divided into three sections. The first part provides a brief background on the Health Information Systems Programme (HISP), its origin and how it works to accomplish its goal. The second part will present DHIS2 before the DHIS2 Design Lab is introduced and described.

2.1 HISP

HISP was initiated as a part of the political processes in post-apartheid South-Africa in 1994. At the time, fragmented and constantly changing health systems and data requirements was a big challenge (Braa & Sahay, n.d.). In effort to establish an integrated and decentralised health system, teams were put together to develop plans for the reconstruction of the health sector. The University of Cape Town, the University of Western Cape and a Norwegian PhD candidate from the University of Oslo (UiO) were engaged and formed a collaborative project to develop a district-based health information system (Sæbø et al., 2011). The strategy of the project was focused on a flexible data structure, involving development of tools, data standardization, datasets and software applications to support implementation (Sæbø et al., 2011) influenced by the Scandinavian approach of Participatory design (PD) and Action Research (Braa et al., 2007). This project was the beginning of HISP and its result was DHIS which proved to be a huge success.

HISP has the overall goal to "*enable and support countries to strengthen their health systems and their capacity to govern their Health Information Systems in a sustainable way to improve the management and delivery of health services*" (UiO, n.d.-c). In doing so, HISP has grown to be an evolving global network of action, research and development (Braa & Sahay, 2012). This has involved facilitation and leveraging of interaction between practice and theory to develop knowledge. Academic research has from the beginning played a significant role, allowing engaged students to contribute with improving DHIS2 (DHIS2, n.d.-b). Faculty researchers, as well as PhD and master students in the DHIS2 ecosystem frequently produce research published in international

journals (DHIS2, n.d.-c) of value to HISP and the Information Systems (IS) research field (UiO, n.d.-c).

2.3 DHIS2

DHIS2 (the second DHIS) is now an open-source, web-based software solution mostly used as a Health Management Information System (HMIS) (DHIS2, n.d.). Today DHIS2 (figure 1) is used by 73 countries worldwide, making it the largest HMIS in the world (DHIS2, n.d.-b). To manage its rapid growth, DHIS2 has taken a platform-approach by developing a generic core software (Li & Nielsen, 2019b), deferring further implementation and configuration to the local contexts. In this thesis, implementation refers to the adaptation of the generic software. Configuration refers to the work of shaping the generic software to the organization's requirements and context. In order for DHIS2 to be implemented and used in different local contexts it has to be flexible and generic enough to accommodate a wide variety of functional requirements, resulting in the need to be implemented and configured in line with the local context (Li & Nielsen, 2019b). Only generic requirements, such as requirements relevant across several organizational contexts, are implemented in the generic core software (DHIS2, n.d.-a). Specific requirements are addressed in the implementation and configuration process.

Figure 1: DHIS2 in use by health workers

The development of the generic core software happens at the University of Oslo, by HISP UiO and its around 30 developers and designers (Li & Nielsen, 2019b), further referred to as *the DHIS2 core team*. HISP UiO also has the leading role in HISP, being the main

organization to coordinate and manage DHIS2 (DHIS2, n.d.-a). *Implementation specialist groups* (ISGs) are the implementation partners in the DHIS2 ecosystem in charge of implementing and configuring the generic software to specific organizational contexts. The DHIS2 ecosystem refers to the processes of software development of the core, implementation and configuration, the actors involved, as well as the relationship between these.

2.2 The DHIS2 Design Lab

The DHIS2 Design Lab is a *generic software design lab* (Li, 2019b) located at HISP UiO. It was established in 2018 with the aim to explore how design can be supported and promoted within DHIS2 to address software usability and local relevance for end-users (UiO, n.d.-a). This focus involves strengthening the processes in two contexts of the DHIS2 ecosystem: when the generic software is developed by the DHIS2 core team, and when it is implemented and configured by the ISGs to provide value to the end-users.

The DHIS2 Design Lab consists of what we refer to as the *Design Lab members*, who are master students at the Department of Informatics at UiO (UiO, n.d.-a). The Design Lab members work on projects oriented around understanding problems, designing artefacts and participating in organizational interventions. Examples include exploring user-oriented design and innovation practices, building a method toolkit for design, and exploring efficient component reuse for DHIS2 web application development (UiO, n.d.-a). Following the tradition of the Action Research approach in HISP, the DHIS2 Design Lab has the overall approach of valuing close interaction between practice and research. Consequently, research and software development is integrally aligned forming a collaborative practice with both practical and theoretical contributions (DHIS2, n.d.-c).

Several Design Lab members currently produce research relating to the call for an increased understanding of how to support the implementation processes of generic software in related research (Dittrich, 2014; Pollock et al., 2007; Sommerville, 2008). There are a few cases of Design Lab members with research topics concerning other enterprise software ecosystems than DHIS2, such as SAP, and non-organization specific

topics such as Covid-19 surveillance. This emphasizes the goal HISP has on producing research relevant to other contexts within the same research field.

The Design Lab members also act as a natural group for conducting field trips. The destination for these field trips is usually the offices of ISGs in for instance India, Tanzania, Uganda, Malawi, and Mozambique. While field trips have been postponed due to the Covid-19 pandemic, it is a central activity in the lab and Design Lab members are encouraged to conduct one or more field trips as a part of their work.

Figure 2 is an illustration of the DHIS2 ecosystem and the relation between the different stakeholders that is a part of the organization.

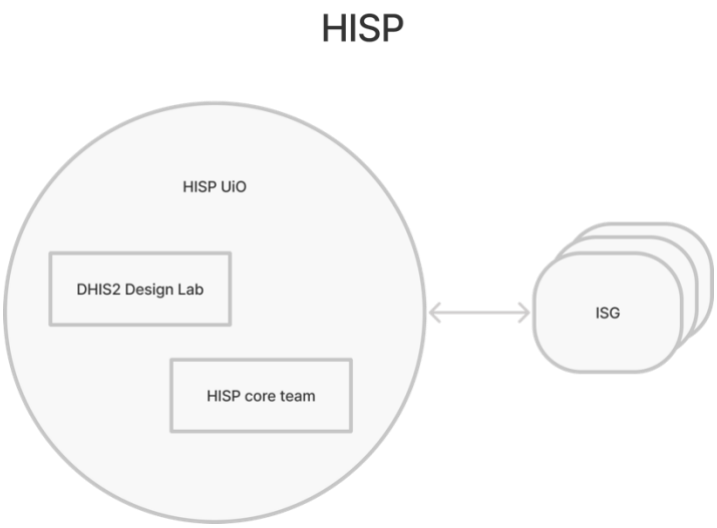

Figure 2: Overview of the DHIS2 ecosystem, HISP UiO, ISGs and their relation.

There are three central stakeholders involved in this particular research project. Table 1 to summarizes who these stakeholders are and how we describe them in this thesis.

Stakeholder	Description
Design Lab members	Master students at UiO exploring how design can be supported and promoted within DHIS2.
The DHIS2 core team	Researchers, developers and managers working directly with the generic core software of DHIS2.
ISGs	Organizations that are implementing and configuring the DHIS2 software to specific use contexts.

Table 1: Stakeholders within HISP

Chapter 3 - Related Research

The main goal in this thesis is to identify and describe what roles a design lab can play to strengthen the software usability within enterprise software ecosystems.

The research is situated between two streams of literature: The first concerns the use of different types of Design Labs as a method to address usability issues. Due to lack of research on using lab-type approaches in the context of enterprise software ecosystems, we identify a gap in the literature. The second stream concerns how working with software usability in enterprise software ecosystems is done, challenges in this work, as well as efforts to address these challenges. Finally, we describe the theoretical lens used in this thesis.

3.1 Working towards usability

The following section describes what usability is and how it relates to the users of the software. Then, we describe four different conceptualizations of labs (referred to as design labs) from the literature. The conceptualizations all value insights and perspectives gained from end-users. The conceptualizations however, differ how these insights and conceptualizations are used to address usability issues. Finally, we provide a summary of how each of the design labs approach usability.

3.1.1 Working with users

Usability is a well-established and acknowledged goal of all types of software development. Usability in our context refers to an objective for systems design which includes enabling users to achieve goals effectively, efficiently and with satisfaction, taking account of the context of use (ISO, 2018). One prominent direction towards achieving usability is through involving users in the development and design of the system. A comprehensive body of research emphasizes the importance of use-oriented design, considering end-users needs and practices (Ehn, 2008; Kujala, 2003; Li, 2019a; Norman, 2013; Rosson & Carroll, 2007). Two traditional directions revolving around

end-users are User-centered design (UCD) (e.g. Gulliksen et al., 2003; Norman, 2013), and PD (e.g. Simonsen & Robertson, 2012). To accomplish usability, the use case, existing practises and mental models of the users should be taken into consideration when the system is designed (Martin et al., 2005; Rosson & Carroll, 2009 in Li & Nielsen, 2019). This argument is also supported by UCD (Gulliksen et al., 2003) and PD (Simonsen & Robertson, 2012). Common for both of these directions is the goal of identifying who the users are, and developing the system with the users' practices and needs in focus.

3.1.2 Usability laboratory

One conceptualization of a lab related to usability in the literature is the *usability laboratory* (Nielsen, 1994). This conceptualization involves inviting end-users to participate in doing tasks based on the intended usage of the software interface. Quantifiable metrics such as number of errors made by the user, time spent, and perceived satisfaction are recorded (Nielsen, 1994). These metrics allow for comparing different software interfaces, selecting the one with the desired results. Additionally, the metrics can highlight specific parts of the interface in which the user makes errors. Recording these metrics requires a certain set-up, with equipment such as screen recorders, two-way mirrors, and cameras. This equipment allows designers to conduct usability tests without directly observing the participants, reducing the chance of getting skewed or biased data. It also ensures thorough documentation of the work, by recording the participant(s), as well as movements and interactions with the system.

This testing environment takes a significant amount of time to set up. By furnishing a dedicated laboratory with this equipment, it allows designers to conduct user tests at a fast pace, minimizing the effort required to set up and conduct the usability test. (Nielsen, 1994). By furnishing a dedicated room with equipment for usability tests, designers are allowed to rapidly test the perceived usability of software by collecting metrics from usability tests done by end-users.

In addition, the laboratory may also serve as a regular workspace to conduct workshops, design sessions, or heuristic evaluation sessions, possibly using the equipment to record

the sessions (Nielsen, 1994). Following this conceptualization, the design laboratory acts as a testing facility, as well as a workspace to work with design.

Bødker and Buur argue that this traditional approach of usability laboratories is "*too limited*", and supports this claim with three arguments (Buur & Bødker, 2000). The first argument is the issue of creating a realistic representation of the real-world contexts. Despite efforts such as the two-way mirror, and recordings of interactions making note-taking and documentation less intrusive, the controlled laboratory environment is not natural, and may not resemble the actual context of use. The second reason is the loss of the ability to get a deeper understanding of the before or after the artefact is used (Buur & Bødker, 2000). The usability laboratory only focuses on the artefact during use, but does not provide insight into what happens before it is used, or what happens after. Looking at the before and after use allows for deeper understanding of which processes and routines that surround the artefact. The third reason is how late the usability testing is done in the development process (Buur & Bødker, 2000). Usability testing is typically done at the later stages of the development process, thus the opportunity of designing together with other team members is lost. Also, usability laboratories do not facilitate addressing usability issues at an earlier stage in the process, making changes much more resource intensive than if they were addressed at an earlier stage. This may also result in changes being omitted due to the resource cost of implementing the changes.

To summarize: the design laboratory achieves usability through providing a physical space that allows for rapid usability testing with users. By testing several software prototypes, metrics allows for comparison and selection of the most usable software prototype. Additionally, the metrics can highlight specific points of issue.

3.1.3 Design collaboratorium

As a resolution to the described issues with usability labs, Buur and Bødker (2009) conceptualize the *design collaboratorium*. Distancing itself from the term laboratory, the design collaboratorium regards the space of the lab as a collaborating space for designers, usability professionals and users, a workplace for design to take place, as well as serving a reflection of the actual use context (Buur & Bødker, 2000). For instance, in one of the

projects Buur & Bødker report from, the purpose was to design a PC situated in a living room, and in the process of doing this the design collaboratorium functioned as a replication of a living room. This replication is argued to encourage the surfacing of tacit knowledge about the context of use (Buur & Bødker, 2000). Following this conceptualization, the design collaboratorium serves two purposes: as a meeting ground usability-related methods and activities to take place, and as a reflection of the use-context.

One can then ask the question of what these usability-related activities are. While the conceptualization of the usability laboratory describes the practical method of achieving usability, the design collaboratorium does not outline these methods:

"The design collaboratorium is now ready to take on its own life as an established usability practice, though not as a static set of methods. Hence, usability professionals need to continuously reflect on methods and process, experiment with new techniques, and in effect develop product and design processes simultaneously." (Bødker & Buur, 2002, p. 168).

The conceptualization rather opens for each instance of the collaboratorium to create their own methods and processes, designing both the software in question, but also its own internal processes. Common for these processes is co-creation and knowledge sharing between members. While the members may work with different tasks, sharing insight of each other's work are central in the conceptualization.

To summarize: the design collaboratorium achieves usability through encouraging tacit knowledge about the usage of the software. This is done by the collaboratorium acting as a reflection of the context of use. Additionally, it encourages the creation of appropriate methods and processes, which fit each particular project.

3.1.4 Design:Lab

To describe collaborations between stakeholders with a mutual interest in design research, influenced by the many different concepts of labs such as learning labs, innovation labs, and usability labs, Binder and Brandt (2008) introduce the expression *Design:Lab*.

Traditionally, the term "laboratory" conjures pictures of experiments, lab-coats, measuring - a picture far from the practices of UCD or PD, even usability in general. Working with usability through a traditional laboratory approach would imply solving usability-related issues through the creation of hypotheses, conducting scientific experiments, rigorously documenting the process with a goal of reproducibility in other labs. However, it is argued that looking at the nature of traditional laboratories as a metaphor is of value, and are useful for design research (Binder & Brandt, 2008).

One aspect of this metaphor is the environment of the Design:Lab. Binder & Brandt (2008) argue this environment is the result of what the participants in the Design:Lab regards as the boundaries of the project, rather than a physical space, office or room. However, there is also a physical aspect: activities or work conducted in a Design:Lab require certain equipment, for instance workspaces, conferencing technology for remote communication, or whiteboards used in workshops.

Another aspect of the metaphor is the methods or processes in the lab. Working with hypotheses, rigorous experiments with a focus on repeatability in laboratories is far from traditional usability work. But, by abstracting the notion of experiments from the traditional to a metaphorical way of thinking of experiments as a study of action and consequence, Binder and Brandt argue the metaphor can be still used in the context of usability studies: "*But if we take a broader view of experiments as something we engage in to discover consequences of actions that interest us, then this may describe what is going on in the Design:Lab*" (Binder & Brandt, 2008, p.119).

This more open way of thinking about processes also allows for process innovation as well. The space should prototype and experiment with different processes, taking advantage of the generative space in the lab (Binder & Brandt, 2008). While working

towards a product or artefact, the processes of work should also be changed and iterated upon, with the possibility of sharing and reuse within as well as outside the lab.

Another aspect of the metaphor of using laboratories for working with usability is record keeping. The Design:Lab is a place to store artefacts or illustrative works resulting from work done (Binder & Brandt, 2008). Displaying these to the members of the lab serves as a reminder of the different directions of work being done, and can also function as a centerpoint for discussions. In a laboratory, detailed reports or articles with documentation of findings are written, which again disconnected from usability work. The goal of ensuring valuable work is not lost, and ensuring iteration on projects is still of great importance. By iterating and innovating on the traditional means of keeping records, value can be gained.

Its strength lies in combining different formats of records with the cooperation between members of the organization (Binder & Brandt, 2008). As stated above, the physical space can also act as a repository or arena to store and display artefacts such as prototypes, graphical elements like posters, maps, and even other resulting material from the work done. As a result, the laboratory becomes a record keeper, a library containing earlier work and results of projects.

To summarize: The Design:Lab works toward usability through the creation of a collaborative space, where design experiments invite interaction and co-creation between all the members. There is a central focus on recording results, ensuring learnings are not lost, and may be iterated on in further experiments, furthering usability.

3.1.5 Living labs

Living Laboratories, in short living labs, is an approach to facilitate co-design between multiple actors such as developers, implementers and future users (Andersen et al., 2018). Like the design collaboratorium and Design:Lab, living labs offer the aspect of exploring elements of the design that is related to the specific use context (Kjeldskov & Skov, 2014), and the transition from design to implementation (Andersen et al., 2018). In practice, living labs are installation set-up in natural, or semi-natural settings over a period of time

to bridge design and real-life use (Kanstrup, 2017). However, living lab is argued to be a broad term used in ambiguous ways (Alavi et al., 2020; Andersen et al., 2018). The notion of living labs has evolved with time since its first introduction in 1999 which calls for a consensus of associated concepts (Alavi et al., 2020).

A recent study by Alavi et al. (2020) identified and described five strands of living labs: Visited Places, Instrumented Places, Instrumented People, Lived-in Places and Innovation Space. *Visited Places* refer to environments resembling living spaces such as a kitchen, living room, or office set up for research purposes. These types of living labs have a fairly controlled test environment that seeks to be a naturalistic environment (for example kitchen, living room, etc.) where people spend time being exposed to experimental conditions. *Instrumented Places* are real living environments where participants spend time at their own homes or offices. Researchers instrument the place with sensors to collect data over a period and can produce outcomes with high levels of ecological validity. Instrumented People refer to a community of people who agree to participate in a study where they carry wearables and/or agree to be recorded with their smartphones. Due to the independence of a physical location, scalability in terms of the number of participants is an advantage in these living labs. Lived-in Places are built as a research facility similarly to Visited Places, but are functional environments used as for instance real apartments, offices and schools. The people who are using the facilities are permanent participants aware of the fact that they are a part of a research project and when, where, how and why they are observed throughout the day. The reason for participating in these research projects can be financial or interest in testing new concepts, services or technologies. Finally, *Innovation Spaces* refer to a social innovation environment where companies, research organizations and individuals come together to work on a technology, co-create and prototype. Users are often involved in the design process early with the goal of evaluating the usability of a technology. With these classifications of living labs, Alavi et al. (2020) invite researchers to adopt and complement them with descriptions of their living lab studies.

Although living labs have received a great deal of attention in HCI literature and as a method, the research is argued to be characterized by recommendations rather than

examinations (Kanstrup, 2017). Hyysalo & Hakkarainen (2014) argue that there is little detailed empirical research of the merits of living labs with a focus on how things *should* happen. Rather, the focus is more toward what *can* or *potentially could* happen. Building on this claim, Hyysalo & Hakkarainen (2014) explored and compared two technology-driven health projects, one which relied on a living lab, and one that did not. Their findings indicate that the living lab did facilitate a quicker resolution to challenges concerning the design process.

To summarize: while there are several strands of differing living labs, there is a common focus on exploring elements of the design in relation to the context of use. The use-context may have severe implications on the usability, which living labs seek to include in the design process. Usability is also achieved through having the user participate in the whole development process. Table 2 contains a summary of how each of the design labs works towards usability.

Design Lab	**Addressing usability**
Usability Laboratory	Dedicated room to do usability testing with users Metrics (such as time, # of errors) are recorded, and used to determine the usability of the software
Design Collaboratorium	Serves as a common space for collaboration throughout the project. Members are aware of each other's work. Encourages tacit knowledge about the use context through replicating the context of use Relevant internal methods and processes should be created on a case-by-case basis, based on what is relevant for each project
Design:Lab	Creating a collaborative space with a common understanding of a common goal for the project and its context of use Conducting design experiments, with the goal of producing, recording, and storing knowledge Innovate its own relevant methods and activities
Living Lab	Exploring elements of the design related to the context of use Focus on involving users throughout the whole development process, from early design to implementation

Table 2: Summary of how design labs work towards usability

3.2 - Enterprise software ecosystems and usability

We now turn to the second stream of literature relevant to the topic of this thesis. First, we describe the generic software within enterprise software ecosystems, as well as how it is implemented to a specific use context. Then, we discuss usability issues within these ecosystems. Finally, we describe one approach of working towards rectifying these usability issues.

3.2.1 Enterprise software ecosystems

Generic software is defined as systems designed and developed for general use and are meant to not be related to one single organization and become specialized, but rather made to be of interest to a market of several organizations (Bansler & Havn, 1994). In the case of introducing such software to an organization, implementation including configuration to the specific use contexts is (in most cases) necessary (Xu & Brinkkemper, 2007). The development process of generic software can be described as a two-step approach with a vendor developing and maintaining the generic software, and implementation partner being in charge of implementation and configuration (Bansler & Havn, 1994).

Generic software can be divided into different types based on their inherent ability to be configured. An early such division was made by Davis, who defined three types of generic software: packages installed with no tailoring, packages installed with pre-specified options for tailoring, and packages installed with custom tailoring (Davis 1988 in Bansler & Havn, 1994). In addition, Davis suggests one additional type of generic software, looking at generic software as development tools, such as programming languages. Dittrich and Vaucouleur (2009) also points to how generic software, specifically Enterprise Resource Planning Systems (ERPs), can have multiple possibilities to be configured, ranging from modules to source-code adjustments.

One common strategy within software development is to build an *enterprise software ecosystem*. This strategy involves that the vendor gives development access to the generic software to implementation partners, which previously were restricted to the vendor (Hanssen, 2012). The term ecosystem describes the processes of generic software

development and maintenance, implementation and configuration, development of custom applications, the actors involved, as well as the relationship between these (Dittrich, 2014).

There are three actors in an enterprise software ecosystem: the vendor, which is the organization owning the generic core software, implementation partners, which develop on top the generic core software, and end-users, the intended users of the software (Hanssen, 2012). In this thesis, the DHIS2 core team acts as the vendor, ISGs as the implementation partners, and the users of the implemented software as end-users.

Evolution is a central topic of generic software as it constantly evolves to answer to the changes in the respective ecosystem (Dittrich, 2014). This is typically changes related to the use contexts of the organizations implementing and configuring the generic software. Dittrich (2014) argues that keeping in contact with the context of use is crucial as it provides the vendor with the main requirements and inspiration for the generic software. The results derived from projects of implementing the generic software are also ideally fed back to the vendor supporting a cycle of innovation and evolution of the generic software. This calls for a need for communication to flow between the vendor and implementation partners (Dittrich, 2014). In order for an enterprise software ecosystem to be successful, all actors and their needs have to be taken into consideration through the process of development and evolution (Dittrich, 2014).

3.2.2 Usability in enterprise software ecosystems

As opposed to bespoke software which can be defined as software made for one particular organization, the process of building generic software requires a well-defined architecture that easily can be adapted to changes in organizational contexts (Xu & Brinkkemper, 2007). Earlier, we have argued that the use case, existing practices, and the end-user's mental models should be taken into consideration when a system is designed. But what if the software is designed to fit several use cases? The established practice of designing for usability is incompatible when designing generic software, which aims to fit general use, rather than specific users (Bansler & Havn, 1994). As articulated by Norman (1998, p.

78): *"making one device try to fit everyone in the world is a sure path toward an unsatisfactory product; it will inevitably provide unnecessary complexity for everyone"*.

Projects where generic software is implemented are argued to fail because the social and organizational complexity of the environment where the software is implemented is not recognized in the development process (Baxter & Sommerville, 2011). As argued in section 3.1.1, it is important to consider the end-users' needs and practices. Having this focus is argued to be of high relevance in terms of software usability, enabling the users to achieve a desired efficiency, effectiveness and user satisfaction (Grudin, 1991; ISO, 2018).

Usability is argued to be a major challenge both when generic software is developed and implemented (Li & Nielsen, 2019b), and creating a working system (from the perspective of the users) are described as an "accomplishment" (Pollock et al., 2007), rather than a given outcome.

The resulting gap between the generic software and user practice of a particular organization is also well documented in the literature (Berente et al., 2016; Sia & Soh, 2007; Strong & Volkoff, 2010). While it is argued that these usability issues can be solved through a thorough process of configuration, it is also emphasized that this activity is not without issues. Several articles have discussed the challenges during the processes of configuring generic software (e.g. Martin et al., 2005; Mousavidin & Silva, 2017; Sommerville, 2008; Strong & Volkoff, 2010). As a result from a deficient process of implementation, several types of misfits or misalignments between system and context have also arisen (Sia & Soh, 2007; Soh et al., 2000; Strong & Volkoff, 2010). In this context, we refer to misfits or misalignments as disconnects between the organizational requirements and the capabilities of the system. There are, however, ways of closing the gap between the "generic" and the "specific".

3.2.3 Addressing usability in enterprise software ecosystems

In addition to the misfits and misalignments between system and organization, scholars have also suggested areas of further research, as well as efforts to overcome the challenges

of implementing generic software. Acknowledging the importance of conducting a thorough process of implementation is central to rectifying misfits, as well as allocating enough time for a proper process (Sia & Soh, 2007). Carrying out this process, however, is often not as straightforward.

From the vendors' perspective, using traditional design methods for working with the implementation would be an impossible task when considering how many use contexts, end-users, and requirements that follow generic software. As argued by Monteiro et al.: *"Design within [enterprise software ecosystems] are not confined to a single time and space, but rather happens over a longer period of time, and at several places*" (Monteiro et al., 2013, p. 575). The traditional approach assumes design happens at one or few places, and is done before use begins when the development project is done. Dittrich argues:

"[..] the software product providers need to take the distribution of design, development and innovation across the ecosystem into account and that the resulting practices are required to extend the perspective of the software engineering methods, tool and technique development beyond the traditional frame of the project." (Dittrich, 2014, p. 1451)

While strategies for rectifying misfits have been established (Mousavidin & Silva, 2017; Soh et al., 2000), there is yet a need for research as well as resources aiding how effectively and efficiently a system can be tailored to a particular context. It is for instance argued that one approach for addressing usability in enterprise software ecosystems is by creating resources aimed at supporting the implementation of generic software (Dittrich, 2014; Ehn, 2008; Li, 2019b; Li & Nielsen, 2019b).

3.3 Theoretical lens: two levels of design

To understand the process of developing the generic core software, developing custom apps and implementing and configuring the core to fit the context of use, we utilize a theoretical lens. This lens consists of three concepts. The first two terms are *generic-level*

design and *implementation-level design* (Li & Nielsen, 2019b). These terms are useful for understanding and describing the processes of designing generic software. The third term is *design infrastructure*. This term is useful for understanding how the DHIS2 Design Lab supports implementation-level design within the DHIS2 ecosystem.

3.3.1 Two levels of design

One term used to describe the design of generic software is *meta-design*, as outlined by Fisher & Giaccardi (2006). Traditional user-centered design assumes the possibility of finishing design before use, and that this process is done solely by designers. Meta-design recognizes that the design process continues after use begins, that users are able to design in cooperation with the developers and designers (Fischer & Giaccardi, 2006). Thus, the users take part in the design process of the software. This is also supported by for instance Sommerville through the concept of *construction by configuration* (CbC) where users are able to configure the system to such a degree that the configured software can be regarded as something new (Sommerville, 2008).

Working from the notion of two levels, Ehn describes two different design processes, *design for use* (for instance PD or UCD), and *design for design* (designing for further configuration) (Ehn, 2008). These divisions are also quite similar to *deferred design*, where different instances of one software are implemented by different actors, with differing use contexts (Dittrich, 2014).

Based on this division between the different types of design, we follow the terms outlined by Li and Nielsen (2019b), *generic-level design* and *implementation-level design.* Implementation-level design refers to the more traditional sense (like PD and UCD) of designing for usability. At this level, the design processes are concerned with a smaller scale, with fewer differences in scope and end-users. This work is usually done by the user organizations in the enterprise software ecosystem, where the goal is to configure the generic core software in such a way to fit the requirements set by the intended context of use, as well as the end-users' existing practices and mental models. There is, however, a major difference between implementation-level design, and the more traditional design,

due to the fact that the configurability of the generic core may set severe limitations on which changes that can be made (Li, 2021).

Generic-level design encompasses the design of the generic software, which possibilities the user organizations actually have to implement the desired changes. Sommerville describes this process: "*the designers of [generic software] systems should recognize that configuring a system is time-consuming and expensive and that the generic system should be designed to simplify the configuration process and to reduce the probability of configuration errors.*" (Sommerville, 2008, p. 9).Thus, it is both a question of the possibility at all of configuring the software, as well as how effective or time consuming this process is.

Li and Nielsen (2019b) also argue that usability is a joint effort between the two levels. If there is a lacking process of generic-level design, the ISGs are unable to configure the software according to the user requirements and feedback. Similarly, having a lacking process of implementation-level design, the generic software is not configured to a satisfactory degree, and does not fit the particular use context. Table 3 from Li and Nielsen (2019b) provides an overview of the relationship between the levels of design.

	For design	For use
Generic-level design	X	X
Implementation-level design		X

Table 3: Generic and implementation-level design in relation to the designing for use, and further design

3.3.2 Supporting implementation-level design

The last term in our theoretical lens is *design infrastructure*. As noted in section 3.2.3, one argued approach to supporting usability is through creating a set of resources, aimed at strengthening implementation-level design. This infrastructure is contributed to, and maintained by the software vendor. Then, during implementation-level design, ISGs utilize these resources, facilitating usability (Li & Nielsen, 2019a).

The design infrastructure may contain different types of resources. Artefacts, such as component libraries or software development kits. Additionally, it may contain training resources, such as online documentation, development tutorials, and design method toolkits (Li & Nielsen, 2019a). Figure 3 provides an overview of the relationship between the terms in the theoretical lens, and table 4 provides a short summary of the terms.

Term in theoretical lens	**Meaning**
Generic-level design	The process of designing generic software, intended to be designed further during implementation. Done by the software vendor.
Implementation-level design	The process of configuring generic software, based on the local context of use. Done by ISGs. Limited by constraints set by generic-level design.
Design infrastructure	A collection of resources, with the aim of supporting usability during implementation-level design.

Table 4: Summary of terms in the theoretical lens

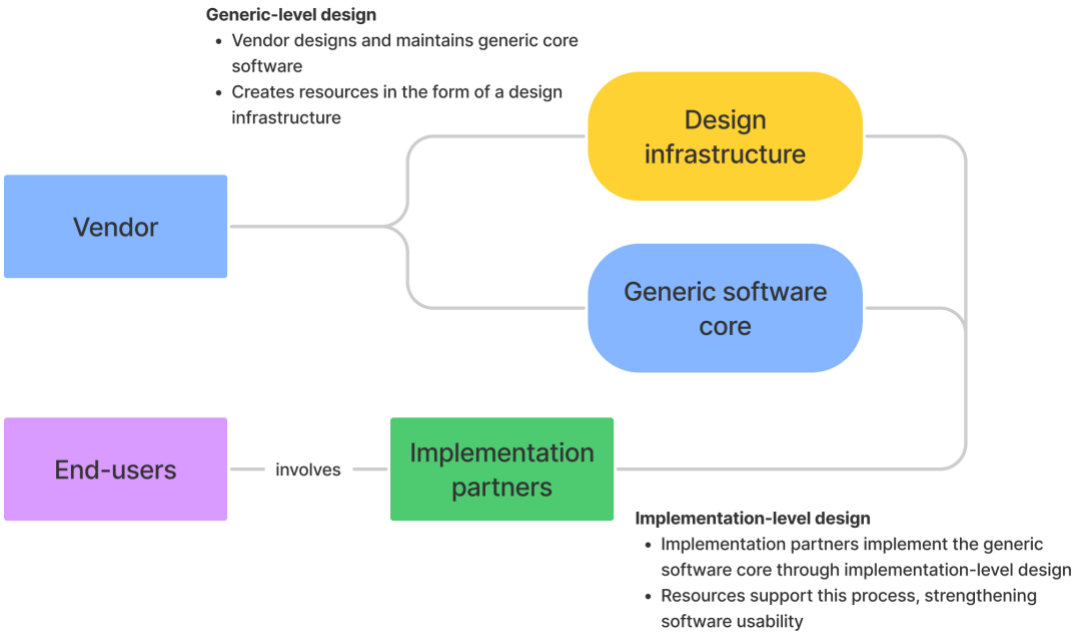

Figure 3: Overview of the relationship between terms in the theoretical lens.

Chapter 4 - Research Approach

The empirical research of this thesis is based on a one-and-a-half-year long embedded case study, including a field trip to Malawi and Mozambique and an engaged partake in the DHIS2 Design Lab at The University of Oslo.

The subject of research for this thesis was initiated in cooperation with HISP and the DHIS2 Design Lab. In the three years the DHIS2 Design Lab has existed, its prominent goal has been to: "*explore how we within the DHIS2 software ecosystem can facilitate and promote the design and innovation of tools that are usable and provide value to the work of end-users"* (UiO). Building on this, our study was initially intended to be an Action Design Research (ADR) project where generating knowledge through interventions would be a focal point. Being an interventionist exploring a relevant research topic within a social environment and essentially learning from this aligns well with the conventional perspective on the mission of IS research: *"make theoretical contributions and assist in solving the current and anticipated problems of practitioners"* (Sein et al., 2011, p.38). However, as the project unfolded, the structure of the project was changed. The methodology, research question and research objectives evolved together with the findings and practical limitations of the project resulting in a research approach we describe as an embedded case study.

To address this thesis research question, we outlined and followed four main objectives: 1) explore the challenges the DHIS2 core team and ISGs face when addressing usability, 2) explore the current practices of the DHIS2 Design Lab and how it works to address usability, 3) explore the potential ways the DHIS2 Design Lab can remedy the usability challenges that the DHIS2 core team and ISGs experience, 4) identify and discuss roles a design lab can play to strengthen the software usability in enterprise software ecosystems based on our empirical case. Each objective has guided the research process, our chosen data collection methods and analysis.

This chapter is organized in the following manner: first we will discuss our ontological and epistemological assumptions, and outline the research paradigm for our study. Secondly, we will describe our research methodology and methods utilized to derive our empirical findings. Lastly, we will describe our approach to analysing the collected data.

4.1 Research Paradigm

Several different paradigms are presented by literature to provide a fundamental philosophical perspective on research. Interpretive, critical, and positivism are three paradigms supporting their underlying ontological and epistemological views presented by Chua (1986) later referred to by several acknowledged works (Klein & Myers, 1999; Orlikowski & Baroudi, 1991; Walsham, 2006). For an interpretive researcher, the usual ontological stance involves being concerned with reality with regard to human interpretations and meanings, and the epistemology stance considers knowledge to be ideological and conductive to particular sets of social ends (Walsham, 1995).

The interpretive research paradigm is frequently used by IS researchers due to the importance of considering social issues related to information systems (Walsham, 1995). Social construction is in many ways the basis of a design lab and the people who are involved, their consciousness and perceptions are what constitutes it. As individuals we have different values, opinions and goals based on our history and previous experiences which impacts our reality and knowledge. It is hard to argue that these evolving, social and subjective entities can be objectively investigated without consideration or awareness of the social factors. The interpretive paradigm supports this view by allowing us as researchers to understand a phenomenon through people's actions, interpretations and meanings (Walsham, 1995). As this resonates well with our research objectives and context, the interpretive paradigm will provide the philosophical perspective for this research.

4.2 Research Methodology: Embedded Case Study

The goal has been to seek knowledge on what roles a design lab can play to strengthen software usability within enterprise software ecosystems by studying a single case, namely the DHIS2 Design Lab. As interpretivists aiming to investigate this evolving, social environment while being directly engaged, we describe our research approach as an embedded case study.

4.2.1 Engaged Scholarship

Engaged scholarship is defined by Van de Ven as "*a participative form of research for obtaining the different perspectives of key stakeholders (researchers, users, clients, sponsors, and practitioners) in studying complex problems*" (2007, p.9). Mathiassen highlights the key characteristic of engaged scholarship that it thus "*draws on the perspectives of key stakeholders in a real-world problem situation and aims to develop knowledge that helps to address it*" (2017, p.19). Being involved as both researchers and members of the DHIS2 Design Lab created a natural close link between the research we conducted in this study, and the "real life" practice within the DHIS2 Design Lab. Establishing and maintaining a close relationship between practice and research is argued to support the creation of relevant research results (Mathiassen, 2002), and has been a focus throughout the research process.

Being engaged in the DHIS2 Design Lab created a close linkage to other relevant instances of the DHIS2 ecosystem as well as the lab itself. This was an important aspect of our study as it allowed us to understand how the DHIS2 Design Lab can remedy the challenges the DHIS2 core team and ISGs face when addressing usability, as formulated in research objective three.

Central to the engaged scholarship approach is the mission of conducting research that is relevant for scientific knowledge while enlightening practice at the same time (Mathiassen & Nielsen, 2008). The contributions from our research are aimed to be of value to both research and practitioners. We contribute specifically to the DHIS2 core team with relevant insights into one approach for addressing the usability challenges the

DHIS2 core team and ISGs face in their work, and generalize our findings to be of value to other similar enterprise software ecosystems as well.

4.2.2 Case Study

Engaged scholarship research can be guided using an abundance of methodologies (Mathiassen, 2017). What however is certain is that the methodology of choice has to be adapted in a manner that will answer the identified problem and research question (Mathiassen, 2017). Case study is a widely used methodology within interpretive IS research (Walsham, 2006). This approach allows researchers to investigate a contemporary phenomenon in its real-life context (Yin, 2003) and is acknowledged to be particularly well suited in the field of IS research (Benbasat et al., 1987). Conducting a case study in an interpretive manner often includes an in-depth look into a case (Walsham, 1995), either to gain a better understanding of the particular case itself, or to illustrate a trait or problem (Stake, 2005).

This thesis aims to answer the research question: *what roles can a design lab play to strengthen the software usability within enterprise software ecosystems?*
The access we had to a generic software design lab was convenient to examine the traits of a design lab in the particular context of interest. Thus, it could provide us with the empirical insights we desired. Case study is therefore a well-suited methodology to address the initial research theme, research question and research objectives of this thesis.

As the research question of this thesis reflects, this study captures a wider theoretical perspective on our topic of interest than an examination of the DHIS2 Design Lab. The DHIS2 Design lab represents other cases and illustrates the topic we are concerned with. Stake (2005) defines this as an *instrumental* approach to case study. Drawing on this term, the notion of generalizability naturally arises due to the results applicability to other contexts. Walsham (1995) discusses generalizations from interpretive research and outlines four types of generalization from interpretive case studies: the development of concepts, the generation of theory, the drawing of specific implications, and the contribution of rich insight. By studying the case of the DHIS2 Design Lab we contribute with rich insight which can be useful to other similar contexts.

4.2.3 Researcher perspective

To answer our research question, we outlined three stakeholders within the DHIS2 ecosystem that were relevant to explore: 1) the Design Lab members, 2) the DHIS2 core team and 3) ISGs. The perspective we as researchers elect to take while exploring the stakeholders shape how we view and interpret the data we collect. To describe our researcher perspective, we draw upon the terms *detached outsider* and *attached insiders* introduced by Van De Ven (2007). Table 5 illustrates our perspective as researchers and relationship with each stakeholder.

Stakeholders	**Researcher perspective**	**Relationship with stakeholder**
Design Lab members	Attached insiders	Collaborative
DHIS2 core team	Detached outsiders	Advisory
ISGs	Detached outsiders	Advisory

Table 5: Researcher perspective

Design Lab members

Given our position as members of the DHIS2 Design Lab, we had a great opportunity to take roles as attached inside researchers while exploring the current practices of the DHIS2 Design Lab and how it works to address usability. It allowed us to continuously gain insight into the case and get an "inside perspective" on work practices such as knowledge sharing, which typically happens in informal settings. If we were to take an "outside perspective", these insights might be lost due to lack of time spent in the environment and access to certain activities. We also took the position as Design Lab Coordinators which allowed us to directly engage in the managing of the lab by planning and facilitating activities. The Design Lab Coordinator position involved planning and arranging for activities with the Design Lab members such as findings presentations and master thesis seminars, to maintain a fruitful research environment and facilitate knowledge sharing.

Being attached inside researchers opens up the possibility to co-produce knowledge with relevant people in the context of interest (Van De Ven, 2007). By taking this collaborative approach to understand the practices within the DHIS2 Design Lab we could gain valuable practical knowledge, and correspondingly our work as researchers could contribute to uncover how the Design Lab members' work could be optimized. This collaborative and engaged approach to research is defined by Mathiassen (2002) as *collaborative practice research*, later used as a classification of engaged scholarship (Mathiassen & Nielsen, 2008). Drawing upon this classification, Mathiassen describes the responsibility of a collaborative researcher as having to "*commit themselves to improving practices and adopt flexible research approaches as practices change and new priorities emerge"* (2002, p.329). Being Design Lab coordinators gave us the opportunity to apply this responsibility while gaining a deep understanding of what the DHIS2 Design Lab is and how it works to address usability.

DHIS2 core team and ISGs

To address the first research objective concerning the exploration of the challenges that the DHIS2 core team and ISGs face when addressing usability, we took the perspective as detached outside researchers. Although we were connected to the DHIS2 core team and ISGs as Design Lab members and as a part of the DHIS2 ecosystem, we did not stand within their respective practice nor had a direct personal stake in the information they provided us with. The implications of taking a detached outside perspective was mainly related to how we were viewed more as outsiders, rather than "one of them". During our field trip to Malawi and Mozambique to visit the two ISGs we had to ask many questions and follow up on topics we were unsure that we had understood correctly. Naturally, language and culture were also a barrier that shaped the circumstances and how we as researchers were approached by the people we interacted with and vice versa. Additionally, we had little information on what challenges the ISGs and the DHIS2 core team experienced beforehand, which emphasized our role as detached outside researchers. In terms of our relationship with the two stakeholders, they played an advisory role of giving feedback to our research activities, while we were in charge of and guided the research process.

Following is a description of the preoperational work that we did prior to the data collection and a description of the methods that we applied in our study, before the data analysis process is described in detail.

4.3 Preparations

Solid preparation is, as most things in life, one of the most important steps towards success. Entering the research context which included an organization with many different stakeholders and many different work practices motivated us to investigate prior insights and experiences from Design Lab members who had already started or completed their research projects. This was particularly helpful in advance of our field trip to Malawi and Mozambique. Walsham points out the importance of doing plenty of homework about a country before traveling to do field research there (2006), and accordingly we did our very best. Conversations and organized knowledge transfer presentations with the DHIS2 Design Lab were useful to prepare us for what we could expect when interacting professionally with people in a foreign country in regard to culture, language and history. One Design Lab member highlighted the importance of thinking critically on how we express ourselves: *"Become aware of what we take for granted in our language and the terms we use".* One example was brought up: *"Who is the «user»? A really important question with many interpretations depending on who you ask".*

In addition, to gain fundamental information on what the work of developing, implementing and configuring generic software consists of, we reviewed published research on the topic. This was done as a supplementary effort to be intellectually prepared to participate in conversations with people we interacted with in Malawi and Mozambique, as well as people in the DHIS2 core team and DHIS2 Design Lab. The readings gave us valuable theoretical insights on concepts such as enterprise software ecosystems, misfits, configuration, design infrastructures, meta-design, usability and local relevance, user involvement, and use-oriented design. Adapting the concepts and having a more common vocabulary was a stepping stone to be able to understand each other and have fruitful discussions.

We found the preparational work to be beneficial to our research, especially in regard to the interpretive approach we took. As we relied on people's interpretations and meanings, we were dependent on being aware of potential learning barriers.

4.4 Methods

We divide our research activities into three categories based on the stakeholders we explored: 1) ISGs in Malawi and Mozambique, 2) DHIS2 core team and 3) Design Lab members. This section is structured accordingly. Table 6 gives an overview of the stakeholders, data collection methods and our aim for the activities reflecting research objective one, two, three and four.

Who	Aim	Data collection activities
ISGs in Malawi and Mozambique	Explore the challenges the ISGs face when addressing usability	Field trip including; informal interviews, workshops, focus groups, demos, presentations
DHIS2 core team	Explore the challenges the DHIS2 core team face when addressing usability	Interviews
Design Lab members	Explore the current practices of the DHIS2 Design Lab and how it works to address usability	Interviews, Lab participation and coordination including: informal interviews, initial lab workshop, standup and morning meetings, presentations, seminars, onboarding workshop, study day, introduction to helpful digital tools
All stakeholders	Explore the potential ways the DHIS2 Design Lab can remedy the usability challenges that the DHIS2 core team and ISGs experience	All activities and methods above

Table 6: Data collection conducted with the stakeholders and the aim for each activity

Four main activities were included in our research process: a field trip to Malawi and Mozambique which included several different inquiries, interviews with the DHIS2 core team, interviews with the Design Lab members, and the lab participation and coordination activity which also included several inquiries. These activities and inquiries are described in the following sections. The data collection process is illustrated in figure 4.

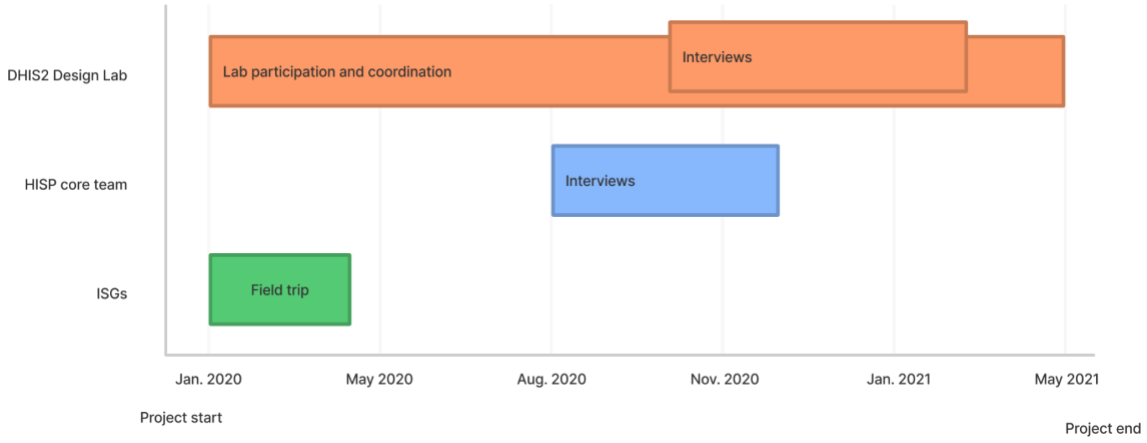

Figure 4: Data collection process

4.4.1 Field trip to visit ISGs in Malawi and Mozambique

In collaboration with two other master students and one PhD student doing research on other topics related to DHIS2 and software usability in enterprise software ecosystems, we went on a field trip to visit two ISGs, one in Malawi and one in Mozambique. The field trip lasted for four weeks between January and February 2020. A prerequisite to understand the ways the DHIS2 Design Lab can remedy the usability challenges that the ISGs (and the DHIS2 core team) face in their work, was to understand what these challenges are. As a consequence of traveling early in the research process, the scope of the gathering of information throughout the trip was quite broad. This allowed us to explore a wide variety of aspects concerning our research topic in the two countries, for instance how they conduct usability-related work, and their relationship with the DHIS2 core team and the DHIS2 Design Lab, as well as other ISGs. We interacted with a number of representatives from the ISGs during our trip, including: developers, implementers, project managers, CEOs, system administrators and coordinators to link the ISG and HISP UiO.

One of the main goals of the field trip was to get first hand understanding of the processes surrounding implementation-level design of DHIS2, especially in regard to how they address usability and any challenges they experience in related to this. Each visit consisted of several activities such as interviews, workshops and focus groups, more or

less in a formal manner. After each day, our notes were discussed and compared with our travel partners.

Informal interviews

Informal interviews, or *ethnographic interviews* (Edwards & Holland, 2013) refer to the more opportunistic and unplanned conversation-style interviews done when we collected data about the challenges the ISGs experience in addressing usability. Not following a particular format or guide, these spontaneous interactions revolve around and are guided by a shared interest or topic. These conversations enrich the insights from other data collection activities, providing a wider picture. This data collection method was applied with one or more ISG representatives at appropriate times during their work-day.

Workshop

At suitable opportunities during the time we spent together with the ISG representatives, we used post-it notes and white boards to discuss interesting topics related to their work (see figure 5). This "active" form of collecting data was appropriate to collaboratively understand for instance how the ISG in focus is connected to the additional DHIS2 ecosystem or how the process of implementation-level design is done.

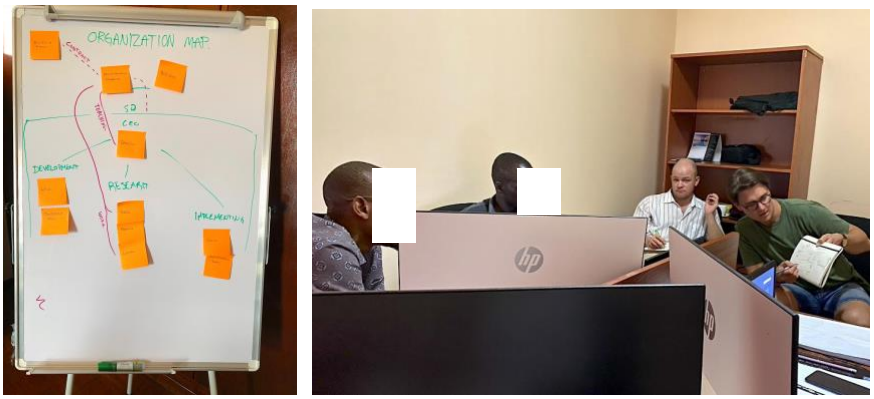

Figure 5: Workshop with ISG representatives

Focus groups

Every visit to the ISGs consisted of group discussions concentrated on specific topics such as project results, communication with collaborating organizations such as Non-Governmental Organizations (NGOs) and the Ministry of Health (MoH), and challenges

with implementation-level design (figure 6). We also discussed their experience of working with users and translating the findings from the usability work into a finished solution. This allowed us to explore how much of an opportunity the ISGs actually have to conduct usability work and whether it is feasible to act upon the findings from this work. These discussions often happened in the context of presentations that we held on findings from previous activities. The goal of having focus groups was to open up for discussions where everyone had the opportunity to speak their mind and share their opinions and experiences.

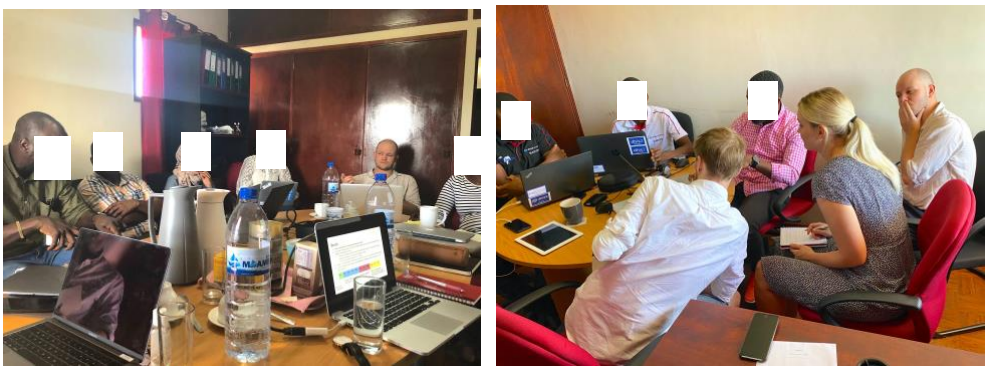

Figure 6: Focus groups with ISG representatives

Demos and presentations

Based on our initiative, demos and presentations were held by the ISG representatives to demonstrate software applications and its functionalities (figure 7). Additionally, presentations were held to present project results and work practices. These activities were an important source of information on what their work consists of and any obstacles they meet in the process of implementing and configuring DHIS2, especially in regard to address usability. These sessions were held by several ISG representatives with different roles giving us a wide perspective on everything from challenges of configuration to project management.

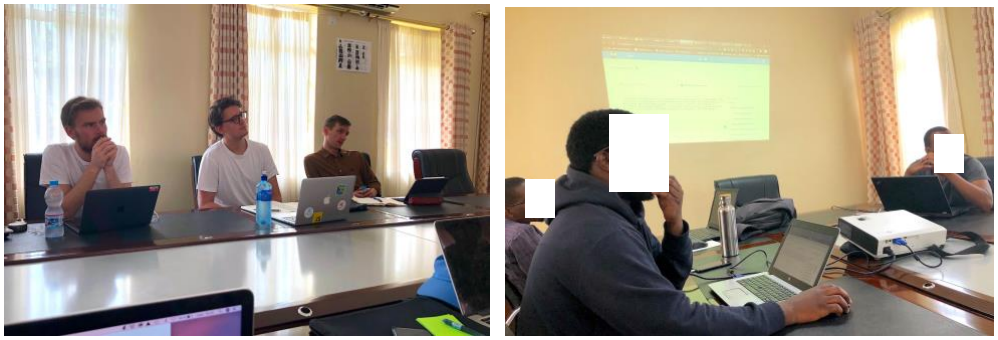

Figure 7: Demos and presentations with ISG representatives

To summarize, table 7 contains a summary of the inquiries for data collection that was conducted during our field trip:

Inquiry	**Description**
Informal interviews	Interviews partly prepared and held when appropriate with representatives from the ISG in Malawi and Mozambique.
Workshop	Sessions of discussions using post-it notes and white boards with ISG representatives.
Focus groups	Group discussions conducted with specific topics in mind, often centered around presentations on our findings from previous activities.
Demos	Walkthrough and demonstrations of software functionality performed by the ISG representatives.
Presentations	Project result and work practice presentations (including implementation-level design) performed by the ISG representatives.

Table 7: Inquiries during the Field trip activity

4.4.2 DHIS2 core team

As a central stakeholder in the DHIS2 ecosystem, the DHIS2 core team could ground our understanding of how DHIS2 Design Lab can remedy usability challenges. We conducted 10 formal interviews with people from the DHIS2 core team, including the DHIS2 core UX lead, Co-founder of DHIS2, the DHIS2 Design Lab lead and Front-end architect.

Interviews

Formal interviews describe the traditional sense of interviews with an aim of gathering qualitative data (Crang & Cook, 2007). This type of interviews follows a semi-structured format and are scheduled with the participants. An interview guide was prepared with concretely formulated open-ended questions with interrogative words such as what, how, and which. By asking open-ended questions, we were able to deviate from and add questions during the session when interesting topics were brought up. It also allowed the participants to interpret the question to their understanding. The main learning goals for these interviews was to understand their role in the DHIS2 ecosystem, their impression of what the lab is and where its value comes from, challenges they identify or experience in addressing software usability in DHIS2, and wishes for how the DHIS2 Design Lab can support the expressed challenges. Table 8 provides an overview of the formal interviews conducted, including the participants role, number of people, and dates.

Participant role	**# of people**	**Date**
DHIS2 core UX lead	1	4/3-2020
DHIS2 Design Lab lead & PhD candidate	1	25/9-2020
Leading position within HISP UiO	1	15/10-2020
Founder of DHIS2	1	21/10-2020
Implementation team members	4	7/10-2020 - 19/11-2020
Project managers	2	2/10-2020 - 5/11-2020

Table 8: Formal interviews with the DHIS2 core team

4.4.3 Design Lab members

To complement the data collection activities with the other stakeholders, we also wanted to gather data from the Design Lab members. The overall goal was to explore the current practices within the DHIS2 Design Lab, how the lab works to address usability, and the potential ways the DHIS2 Design Lab can remedy the usability challenges that the DHIS2 core team and ISGs experience.

Interviews

Similarly, to the formal interviews we conducted with the DHIS2 core team, this method was also applied with Design Lab members. Our main learning goals for this activity was to understand their motivation of being a part of the lab, their interpretation of what the lab currently is and where its value comes from, how cooperation with stakeholders in the DHIS2 ecosystem is done, and wishes for the lab in the future. By understanding this, we could address research objective three: explore the current practices of the DHIS2 Design Lab and how it works to address usability. Five interviews were conducted in total. Table 9 gives an overview of the formal interview conducted with Design Lab members, including their role, number of people and date.

Participant role	**Number of people**	**Date**
Master student 4th year	2	29/1-2021 - 4/2-2021
Master student 5th year	3	28/1-2021 - 1/2-2021

Table 9: Formal interviews with Design Lab members

Lab participation and coordination

Our engagement in the DHIS2 Design Lab as both members, coordinators and researchers defined the activity we describe as lab participation and coordination. Being engaged in the lab as attached inside researchers gave us the opportunity to both participate, and arrange for different activities which has informed our research considerably. The aim with this activity was to understand what the current practices of the DHIS2 Design Lab

is, and the ongoing efforts put in place to address usability through an attached insider perspective (Van De Ven, 2007).

Also of importance, was our aim to explore any wishes and needs the Design Lab members have for their research work. These types of insights were important to develop an understanding of in which ways the work done by the members in the lab could be strengthened. Most of the inquiries were in fact conducted based on ideas and wishes from the Design Lab members. By looking at data collection as a process of internal innovation, we were aiming to eventually identify new ways the lab can remedy usability challenges in collaboration with the Design Lab members. The lab participation and coordination activity has been an overarching effort throughout the research process and has included several inquiries.

Initial lab workshop

To establish a basic understanding of what the Design Lab members wanted and needed from the lab we arranged and facilitated a workshop together with eight members of the DHIS2 Design Lab. The aim of the workshop was to explore and discuss together with the Design Lab members:

1) How they would define the Design Lab
2) What their goal for being a part of the Design Lab is
3) What value can the lab provide to them
4) How can the Design Lab meet these needs

The workshop was structured into three segments. Each segment consisted of a group discussion exercise using post-it notes to organize thoughts and ideas. After every segment, the post-it notes were put up on a whiteboard and presented and discussed further (see figure 8).

Figure 8: Initial lab workshop

Following the workshop, we gathered the post-it notes and wrote a report based on the findings made available for all the Design Lab members. We also presented the findings on a Zoom-meeting for the participant two weeks later which gave us the opportunity to establish a common understanding of what the lab is and what is needed from the lab. We will argue that this workshop was valuable in providing a clarification of expectations, both in terms of what to expect from each other, and what to expect from the lab.

Informal interviews

As attached inside researchers in the DHIS2 Design Lab, spending time there physically and conducting informal interviews with members was a natural source of insights to our research. Correspondingly to the informal conversations conducted on our Field trip to Malawi and Mozambique, this method does not follow a particular format, but is rather guided by shared interests or topics in a spontaneous conversation-style manner (Edwards & Holland, 2013). Clarifying missing or unclear information or picking up on interesting ideas and wishes for the lab are examples of the relevance of conducting these types of interviews. This method was also valuable to supplement the more formal methods conducted through the research process.

Standup and morning meetings

Several measures were made by us as Design Lab coordinators in effort to meet the expectations and needs of the Design Lab members, and for the lab to keep working towards its goal. Based on the expressed wishes from the Design Lab members on continuously sharing and receiving knowledge and resources with the rest of the

members, and getting assistance on their master theses, we arranged weekly standup meetings. Standup meetings are a growing practice within all businesses and industries. Just as the title suggests, it is a quick meeting where all participants stand up throughout the whole session. Everyone got two minutes each to answer three questions: what have you accomplished since the last Design-Lab meeting, what will you be working on until the next meeting, and what obstacles, if any, are impeding your progress. The standup meetings were sometimes combined with a social breakfast to create an informal arena to keep communication flowing. As a result from this inquiry, we could derive insights on the current practices of the lab over a period of time and uncover any evolving challenges and opportunities.

Once the Covid-19 pandemic hit, the physical space was moved online, and we continued with the standup meetings digitally. Following the master students' increasingly pressured study life, we arranged for additional morning meetings in the times when society was locked down. Motivation was clearly expressed as lacking during most of the time of having home office, and to prevent any setbacks in the Design Lab members' study progression, these meetings were mostly a motivational effort.

Presentations and seminars

As the DHIS2 Design Lab continues to grow, so does the need for communication, cooperation and exchange of information between the members. All Design Lab members have their respective research projects contributing to the DHIS2 ecosystem and the IS research field. In their work, the Design Lab members might come across interesting articles, insights, or ideas, which can be of great aid to other members in the lab. Being attached inside researchers in a collaborative environment we took upon ourselves the responsibility of improving the practices we were a part of together with the Design Lab members (Mathiassen, 2002). To accomplish this, we facilitated and participated in several presentations and seminars (see figure 9). Examples of the presentations and seminars we facilitated and participated in are: field trip presentations, literature search and review seminar, read and review seminar, master thesis content seminars and data collection learning objectives seminars. These activities were resourceful to maintain a

fruitful research community and continue to explore the current practices of the DHIS2 Design Lab.

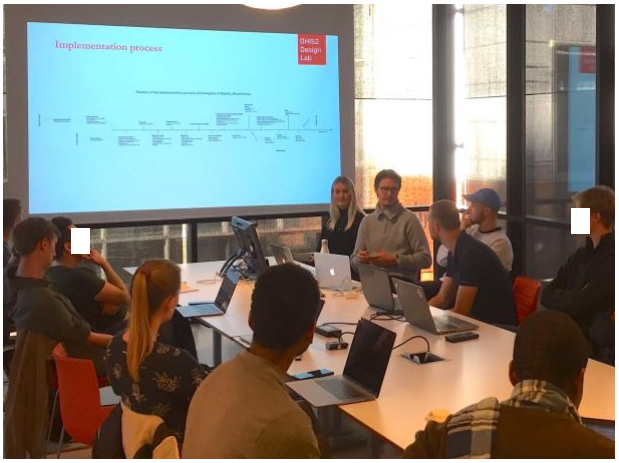

Figure 9: Presentation with the Design Lab members

Onboarding workshop

To follow up on the initial workshop mentioned earlier in this chapter, one more workshop was conducted to brainstorm new ideas, events and activities that could be of relevance to the DHIS2 Design Lab and its members. The workshop was held as a part of the onboarding of new master students and served as a foundation for the recurring activities that would happen throughout the semester. Using the digital collaboration tool, Miro, groups of 4-5 people discussed, wrote down and voted on ideas before they were presented and further discussed in plenary. As opposed to the last workshop, this was held digitally. 23 Design Lab members were present, including ourselves. Our participation gave us the opportunity to uncover any change of interest in the practice of the DHIS2 Design Lab, and stay informed on the current wishes for the lab. In addition, this was also a great opportunity to brainstorm how the lab can strengthen the software usability in the DHIS2 ecosystem through their research contributions.

Study day

To facilitate a fruitful research environment with a practice of continuous knowledge sharing which was expressed as a wish from the Design Lab members, we arranged what we refer to as study days. Study days were days where Design Lab members came in and

spent time together reading, writing and discussing everything that could be of value to their research. We could clearly see that working together was encouraging and a motivational factor for the master students, which resulted in this being an activity we arranged for every week for the time the University of Oslo allowed it (due to the Covid-19 pandemic). Gaining first-hand experience on how the Design Lab members work enriched our understanding of the practices in the DHIS2 Design Lab. This was also an arena for us to passively acquire information on how their research work is going, including any challenges and opportunities they experience.

To summarize, table 10 presents the inquiries conducted as a part of the lab participation and coordination activity, including a description of the inquiry and whether we were facilitators, participants or both.

Inquiry	Description	Facilitators or participants
Initial lab workshop	Workshop to explore and discuss together with the Design Lab members: how they would define the Design Lab, what their goal for being a part of the Design Lab is, what value can the lab provide to them and how the Design Lab can meet these needs	Facilitators
Informal interviews	Interviews clarifying missing or unclear information, picking up on interesting ideas and wishes. Held when appropriate	Participants
Standup and morning meetings	Motivational effort to keep up the study progression, share/receive interesting learnings and to ask/receive any assistance	Both
Presentations	Presentations focusing on exchanging knowledge between the Design Lab members: field trip learnings presentations and Design Lab onboarding	Both
Seminars	Various types of seminars to maintain a fruitful research community: recurring read and review seminars, literature search and review seminar, learning objectives seminar, and master thesis content seminars.	Both
Onboarding workshop	Digital brainstorming session in groups of 4-5 people where the Design Lab members explored ideas, events, activities related to the lab. This provided the foundation for the recurring activities/events that would happen throughout the semester.	Participants
Study day	All Design Lab members gathered in the same room to discuss, read and write, either together or independently.	Both

Table 10: Inquiries during the lab participation and coordination activity

4.5 Data analysis

Data analysis is an important aspect of any research process. How we take apart, analyse and interpret the data we have collected serves as a foundation to how it eventually is presented. Walsham (2006, p. 320) quotes Geertz (1973, p. 9) on how we can describe data in interpretive studies: *"What we call our data are really our own constructions of other people's constructions of what they and their compatriots are up to".* Documentation and discussions have been central when these processes have been taking place in this study. Being two researchers working together and collaborating with other researchers interested in the same topics has strengthened our minds and given us a richer picture. This is also pointed out by Walsham (2006, p. 325) as a benefit: *"I believe that the researcher's best tool for analysis is his or her own mind, supplemented by the minds of others when work and ideas are exposed to them"*.

Research and theory have had a rather complementary relationship in our study. Throughout the whole research process, we have read up on related research concerning our topic of interest. Examples here are challenges with addressing usability during generic and implementation-level design, Design Labs as a method to address usability issues, and enterprise software ecosystems. Thus, theory and existing concepts have played a role in what we discussed and looked for in the data analysis process. Previously reported challenges in addressing usability in enterprise software ecosystems is an example of one theory we found compelling and something we wanted to look for. However, as we were concerned with an unexplored phenomenon which is a generic software design lab and set out to understand how the DHIS2 Design Lab can play a role in strengthening the software usability in its respective ecosystem, our approach has been rather inductive. Empirical data has guided our research process, resulting in a conceptualization of a generic software design lab. Employing research and theory together gave us a more complete understanding of the topic we were researching.

The road from doing data collection to identifying and presenting our findings, to what we ultimately present as roles a design lab can play to strengthen the software usability in enterprise software ecosystems has consisted of several steps within what we describe

as a two-level approach. The first level of analysis happened through reflections and discussions while recording and documenting the data. The second level consisted of understanding the collected data by dismantling, reviewing, discussing, and uncover patterns in it. These two levels are closely interlinked which makes it difficult to state when the process of analysis begins (Stake, 2005). Data collection and analysis happened in parallel forming a research process of cycles illustrated in figure 10. The following section aims to describe the two levels of analysis by first presenting how the data was recorded, before the process of understanding the data through developing codes and themes, and co-analysis is presented.

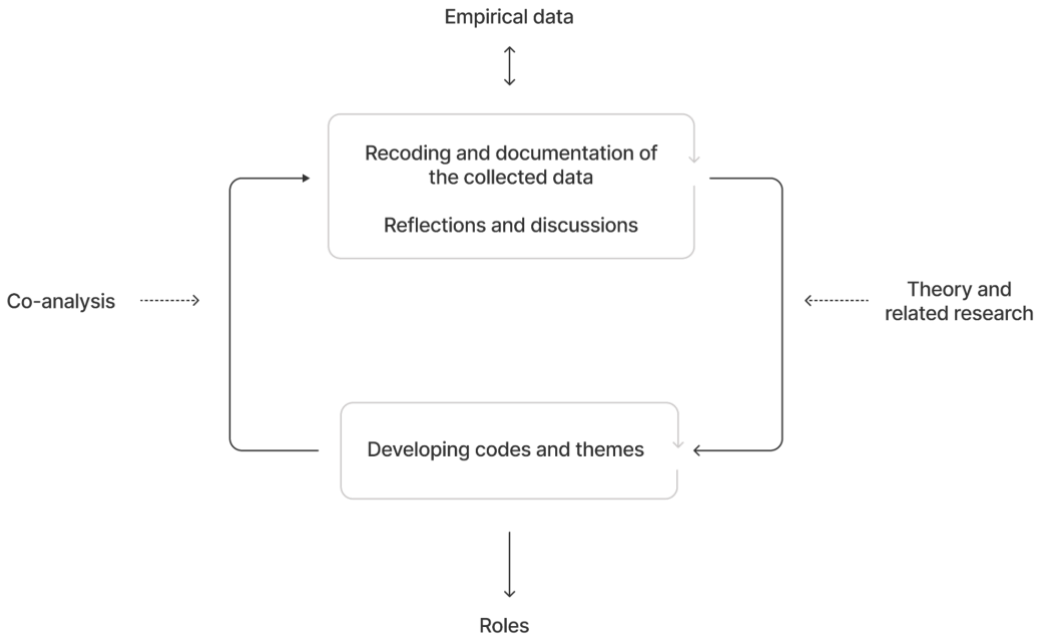

Figure 10: The analytical process

4.5.1 Recording and documentation of the collected data

Through the time we spent collecting data we have been taking notes and recorded quotes, ideas, process figures, abbreviations or anything else of interest to our research. Recording and documenting the research project is crucial to any research as it enriches the process of understanding. After every day of collecting data, we compared and discussed our notes, before we noted down additional text in more detail describing what

we had heard or seen. Being two researchers was especially advantageous at this point in the process as our notes and interpretations complemented each other.

During our field trip we also kept research journals (see figure 11) where we wrote down our experiences, feelings and interpretations every day. Pictures were also taken throughout the days to help us recall any contextual elements such as who was present and at what time the inquiries took place. Additionally, the pictures were also helpful to capture any diagrams or figures we were presented to. As a measure to foster interactive and informative dialogues, we audio recorded all the formal interviews that we conducted. By recording the interviews, we could concentrate on asking constructive questions, rather than be too absorbed in taking notes. All notes, field journals, pictures and audio recordings allowed us to look back on what we found and further guide our research and analysis process.

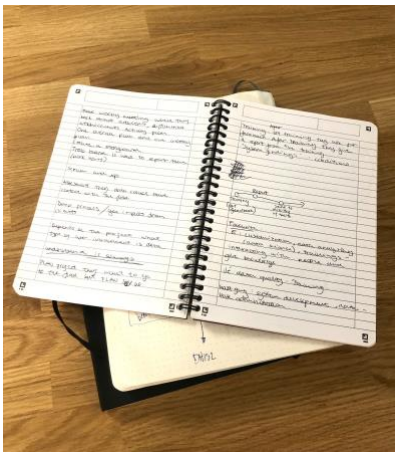

Figure 11: Notes and field journals

4.5.4 Developing codes and themes

Our data analysis is based on thematic analysis. Thematic analysis is an approach to analyse qualitative data aiming to identify, analyse and report patterns or "themes" in the data (Braun & Clarke, 2006). The process of thematic analysis is described by Braun and Clarke (2006) through six stages: 1) Familiarizing yourself with your data, 2) Generating initial codes, 3) Search for themes, 4) Reviewing themes, 5) Defining and naming themes,

6) Producing the report. Thematic analysis was applied in our study as an effort to analyse the initial lab workshop, our field notes from our trip to visit ISGs in Malawi and Mozambique and the formal interviews conducted with Design Lab members and the DHIS2 core team (see figure 12).

In the first stage in the analysis process, we reread our notes and wrote everything down in key words on post-it notes (we continued with the post-it notes that were co-produced with the Design Lab members during the initial lab workshop). We then looked for interesting features of the data systematically in stage two before we gathered all post-it notes and divided them into themes as part of stage three. Examples of themes are: challenge in addressing usability, work tasks and interests, wishes for the lab, and the organization and the lab today.

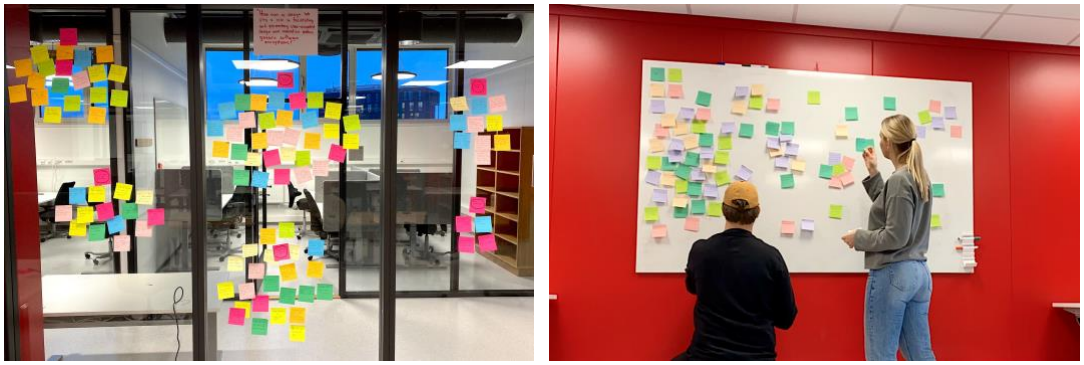

Figure 12: Thematic analysis

Several iterations between stage three and four were made to make sure that we were on a productive track to answer our research question. Lastly, we finalized the themes and outlined the roles presented and discussed in Chapter 6. Table 11 illustrates how we went from empirical data, through codes, theme, to role.

Role	Theme	Code	Quote
Reporting experiences from implementation-level design to vendor	Challenge in addressing usability	Understanding the process of implementation-level design	"*If the same box is moved in almost all counties then maybe the box can be moved in the core*"
		Lack of communication	"*We do not know what the implementation teams need*"

Table 11: Example of role emerging from empirical data

4.5.3 Co-analysis

Working collaboratively, and closely with fellow Design Lab members both in the DHIS2 Design Lab and on our field trip to visit the ISGs in Malawi and Mozambique offered the opportunity to co-analyse the collected data. We all apprehend situations and data differently, and by taking advantage of this we can get a richer picture and collectively build knowledge. It can also allow us to pick up on particularities that others might have overlooked. After every day of visiting an ISG in Malawi or Mozambique we sat down together with our travel companions and shared our notes, discussed them and noted down any new insight that arose. Throughout the research process our collected data and findings was also discussed with other Design Lab members. We will argue that these analysis sessions gave us a more in-depth understanding than we would be able to gain alone.

4.6 Chapter Summary

To summarize, the ontological and epistemological assumptions for this study is based on the interpretive research paradigm. The methodology of our research is defined as an embedded case study with principles adapted from engaged scholarship defining the "embedded". Guided by the research question and objectives, several methods for data collection have provided us with rich empirical insights concerning the Design Lab members, the DHIS2 core team and ISGs. The process of data analysis consisted of several steps within a two-level approach where the first stage was through reflections and discussions while recording and documenting the data, and the second stage aimed to give meaning to the data through dismantling, reviewing, discussing, and uncovering patterns in it. All the methods for data analysis applied in this process informed our research resulting in what we identify and discuss as roles a design lab can play to strengthen the software usability in enterprise software ecosystems. These roles are described and discussed in Chapter 6. In the following chapter, the findings from our research process are presented.

Chapter 5 - Findings

In chapter 1, to answer the research question, we outlined four research objectives. This chapter addresses the first three objectives. The first part provides a more thorough look at the DHIS2 ecosystem, the DHIS2 Design Lab, and how the Design Lab contributes back to the ecosystem. The second part gives an overview of the challenges of addressing usability in DHIS2, based upon empirical findings from the DHIS2 core team, ISGs, and Design Lab members. The third part describes opportunities the Design Lab currently is exploring, as well as future possibilities in which the Design Lab can remedy the aforementioned usability challenges. The final research objective is addressed in chapter 7, where a set of roles are identified and discussed.

5.1 Challenges for addressing usability in DHIS2

This chapter presents our findings regarding the challenges for addressing software usability in DHIS2, based on insights from the DHIS2 core team, the ISGs, and Design Lab members.

5.1.1 Challenges during implementation-level design

Like other generic software, the DHIS2 core team faces the challenge of creating a software generic enough to be relevant for diverse selection of use contexts, yet be flexible enough to be implemented into any particular context of use.

During implementation-level design, the ISGs have two approaches to creating a new solution. One approach involves configuring a part of the generic software. This approach is relatively fast and requires few resources. The configurability of the generic software is directly tied to the generic-level design, and whether or not there are options to configuring the required parts of the software. Lack of configurability may lead directly to usability-problems in the software. Figure 13 shows an example of an interface configured during implementation-level design.

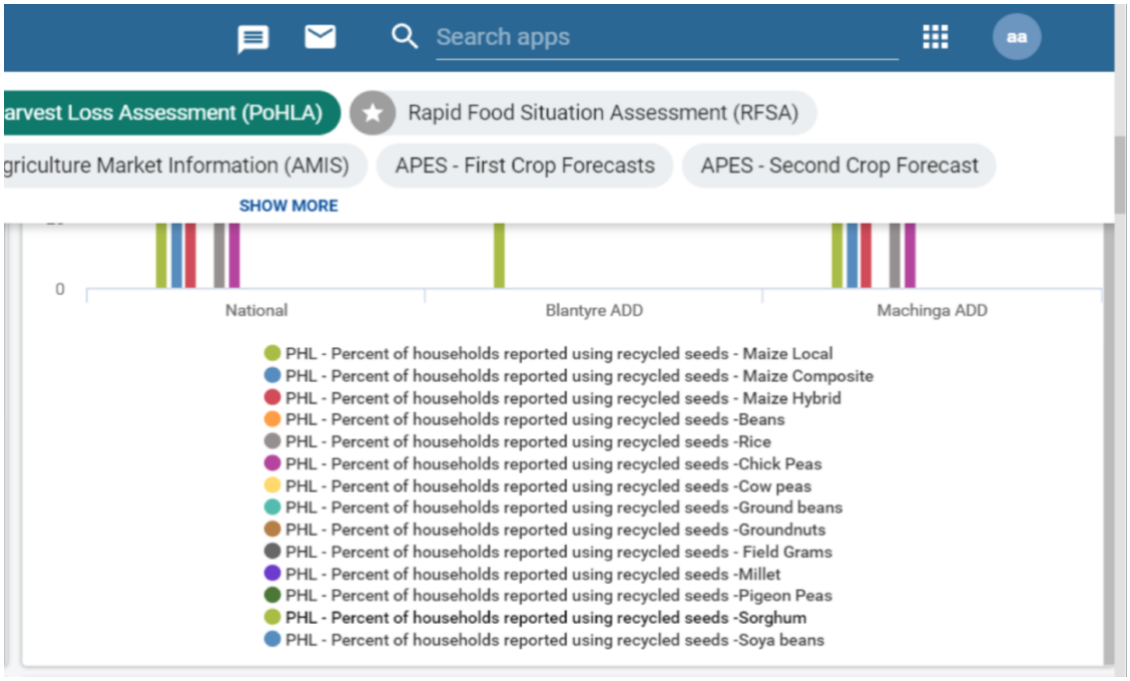

Figure 13: Cluttered and confusing DHIS2 interface

According to feedback from end-users, the graph legend should only show the crop name, such as maize, beans, rice, etc. However, the current interface shows the whole name of the data point. Due to lack of configurability, there was no option for the ISGs to edit the legend. The result is a confusing and cluttered interface, directly resulting in reduced software usability.

As an effort to fix these issues, there are possible work-arounds. Figure 14 shows an example of one such work-around. One of the requirements to the interface was to have the list of data elements sorted in a particular order. However, there are no options for configuring a custom order of these data elements in DHIS2. Thus, the names of the data elements have been given extra characters, to make them appear in a set order when the list is sorted alphabetically. For example, in the first data element, the characters “1 A AR 1 A” have nothing to do with the name of the data element; they only exist to achieve a specific order in the list.

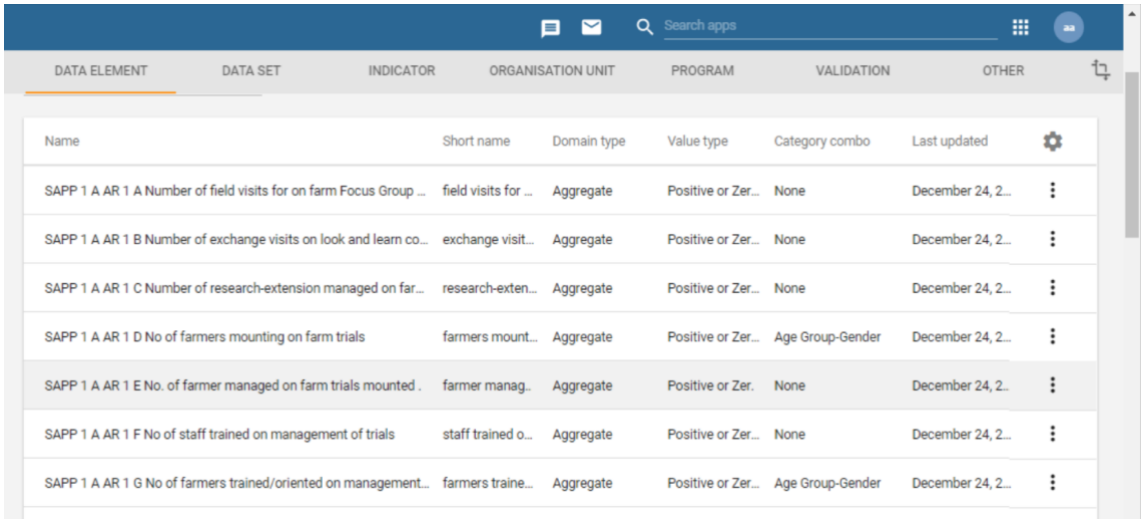

Figure 14: Workarounds for limitations set by configuration options

Similar to the example in figure 13, this leads to an unnecessarily cluttered interface, reducing the usability of the interface. Figure 15 shows a similar example of a workaround. The digits such as 2.1.1 and 2.1.2 are added to get custom sorting in the menu, resulting in a cluttered interface and loss of usability.

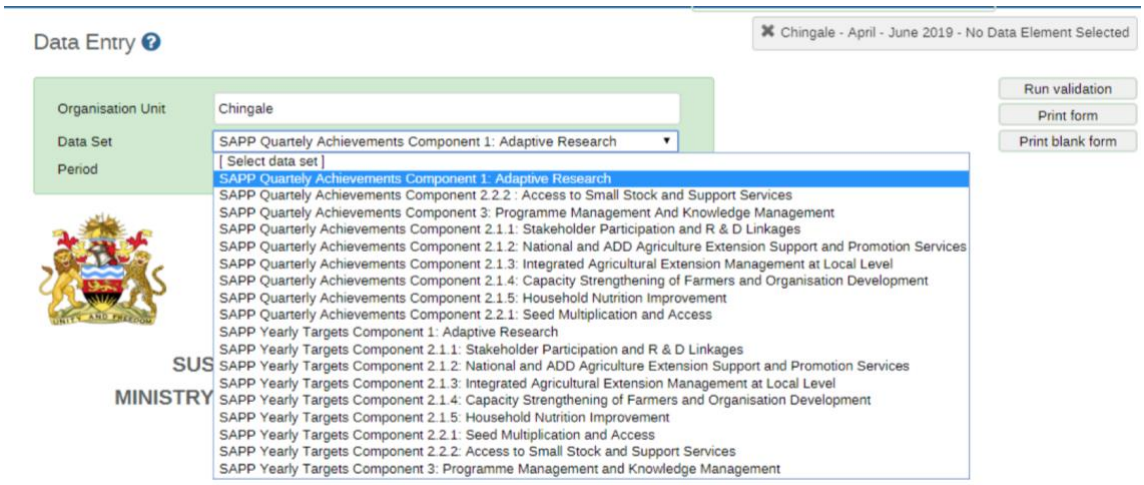

Figure 15: Workarounds for limitations set by configuration options

In addition to creating such workarounds, the ISGs may also submit a request for these possibilities of configuration to the DHIS2 core team. Then, the DHIS2 core team has to review the requests, and implement the changes. The changes will then take effect in a global update to the generic core software. This process takes too much time to be a

realistic solution for the ISGs, which results in either workarounds, or usability issues as illustrated in figure 13. The DHIS2 core team agrees implementing such changes to the generic core software consumes too much time. Discussing how the scale of DHIS2 creates the need of making changes to the software with one of the co-founders of DHIS, we were told "*getting changes into the roadmap is too slow, we need a faster way to make and act upon the changes*".

One other approach to implementation-level design is custom app development. This involves developing a custom application similar to traditional bespoke app development and connecting this app to the generic core software. When the co-founder of DHIS2 was asked to give his opinion on how the issue of these changes taking too long can be solved, the answer was "*supporting the custom app development".* Custom app development allows the ISGs full flexibility in the configuration process, eliminating constraints set by DHIS2. If the solutions in the above examples (figure 13, 14 and 15) were created as custom apps, the ISGs would have the opportunity to do these configurations themselves, removing the need of having the DHIS2 core team take part in the changes.

However, while configuring DHIS2 is relatively fast and easy to do, custom app development requires significantly more time and competence. A central reason for this need is the fact that the development tools originally were built for the DHIS2 core team in Oslo, and not for the ISGs. There exists a trade-off between configuring the generic core software and developing custom apps: spend an increased amount of time and resources in creating a custom solution, where implementation-level design is easier, or to go the cheaper and easier "quick-and-dirty" route, which are more prone to usability-issues.

The importance of supporting custom app development processes was also supported by multiple people central to the DHIS2 core team: "*We need to let everyone build their own stuff*", and *"I believe in shifting the focus towards grassroot app development".* However, when discussing these processes with ISGs, they indicated a need for more resources aimed at supporting both processes. For instance, development documentation and questions from developer fora have been mentioned as often used during custom app

development. In addition, more specialized and technical DHIS2-documentation is also in great need. By strengthening the competence of the ISGs, custom app development becomes a more realistic option of doing implementation-level design.

The ISGs recognize the value of software usability, but are often limited to configuring the generic software due to limited time and competence. The same needs are expressed by the DHIS2 core team. As told by a senior implementation team member: *"we need to provide help or tools to aid implementation"*. It seems there is a definitive need for this infrastructure, from both a development- and implementation-perspective.

5.1.2 Politics, mandates, and contracts limit design focus

Looking at implementation-level design in practice, we found the ISG had a surprising interest in utilizing methods and processes from Oslo. Implementation-level design follows a typical "Scandinavian" approach, iterating on the solution several times, testing the prototype, then going back to earlier steps. One of the issues when discussing prototyping and evaluation, however, was how great the opportunity to conduct these activities is in a project.

Typically, the ISGs do projects on behalf of a project owner such as an NGO or MoH. The ISG then has to negotiate for resources and time to be set aside to be used on user involvement. The project owners typically do not recognize the need for such activities and may also be reluctant to do changes based solely on user feedback. Despite the ISG expressing a need to involve users, the client often declines.

User involvement typically happens at initial stages, often through inception reports, or at the end for verification, implementation, training and handoff. Figure 16 shows the end of a project timeline developed with the ISG in Mozambique. Users are included in many of the steps, but this is typically at the end of the project and not part of medium-fidelity prototyping due to time/resource constraints.

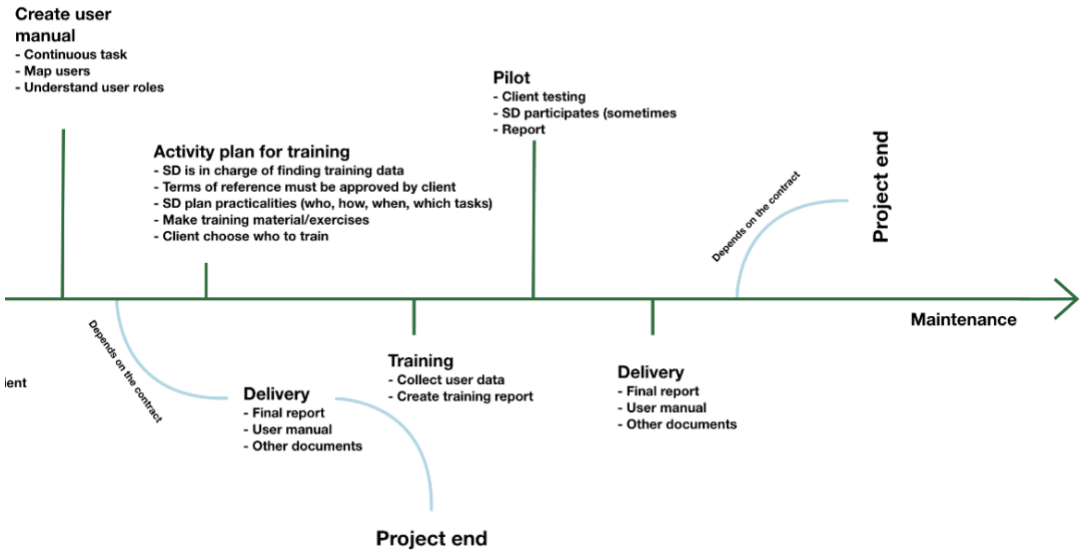

Figure 16: Part of project timeline during implementation-level design

The ISGs is often also responsible for user training, where user feedback often surfaces (see figure 17). They are often unable to act upon this due to the system already being finished, or that the clients are responsible for acting on the feedback. At this point, the ISG has no control whether or not this is done. As a result, the ISGs expressed a clear need for resources to aid in this process, to create opportunity for user involvement in the negotiation processes.

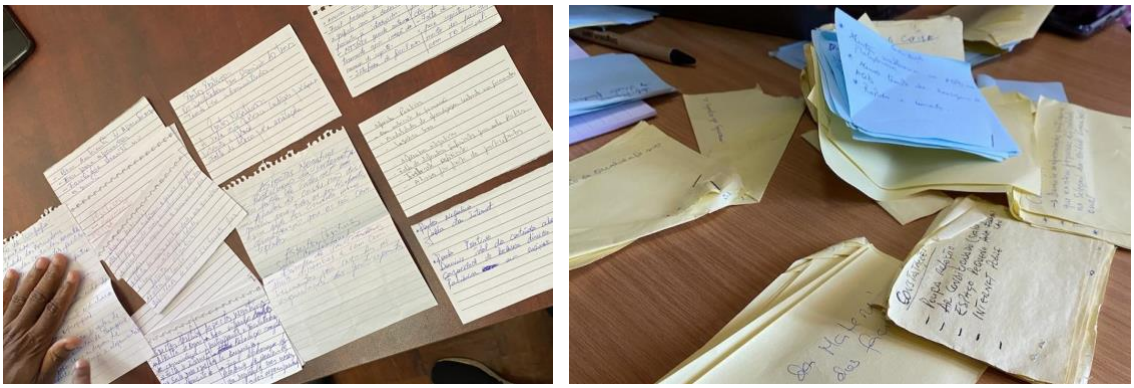

Figure 17: Stacks of notes with feedback from users during training

Having users included in the start and middle of implementation-level design is important, as there is greater opportunity to do changes based on the insights. At the later project

stages, most of the aspects of design are set, and changing these are expensive resource- and time-wise.

5.1.3 Communication with the ISGs

The DHIS2 core team in Oslo faces a knowledge gap between themselves and the ISGs, for instance the challenge of knowing what the ISGs need from the design infrastructure. A senior DHIS2 core team member said: "*There is a central concern in HISP [...] We're losing contact with the local districts*". It was also pointed out in one of the interviews by a front-end architect that there is a challenge of knowing which resources should be created: *"We do not know what the implementation groups [ISGs] need*" and that they are missing feedback on existing resources, both related to development of local apps, and implementation of DHIS2. One such example is one of the development frameworks provided by the DHIS2 core team, which was described as cumbersome: requests to the DHIS2 core can be long and complicated, and prone to errors. Communicating such issues to the DHIS2 core team is important for informing the process of generic-level design.

To summarize: As mentioned in section 5.5.1, supporting the ISGs during implementation-level design is important. If the ISGs do implementation-level design by configuring the generic core software, then any constraints when configuring should be communicated to the DHIS2 core team. During the development of custom applications as part of implementation-level design, the resources in the design infrastructure are especially important, due to the comparatively (compared to configuring) high cost in time and competence. The challenges presented in section 5.1 are summarized in table 12.

Usability-related challenge	Description
Limited design flexibility during implementation-level design	Configuring DHIS2 often meets constraints due to lack of flexibility, resulting in lacking usability
Creating custom apps during implementation-level design is time -and resource-intensive	Developing local apps with DHIS2 is a time intensive process and requires more competence, limiting design focus
Factors are limiting the ISGs design focus	Design-related opportunities are limited by contracts and mandates. Project owner often do not recognize the importance of usability and design
ISG communication	Communicating issues during implementation-level design is important to rectify the issues The DHIS2 core team needs more feedback on existing resources, which challenges ISGs are facing, and which new resources are needed to support implementation-level design

Table 12: Summary of challenges

5.2 The DHIS2 Design Lab and its position in the DHIS2 ecosystem

This section will, based on our findings, present a more thorough perspective of the DHIS2 ecosystem, and how the DHIS2 Design Lab is positioned and how it makes its contributions within it. We will exemplify this position and the contributions made using currently ongoing projects in the Design Lab.

5.2.1 The DHIS2 ecosystem

The DHIS2 ecosystem has two groups of actors in its ecosystem. The first is the DHIS2 core team, which acts as the vendor of DHIS2. Their responsibility includes the maintenance and development of core functionality. They are also tasked with maintenance and creation of resources for the aforementioned processes of implementing DHIS2, as well as developing local apps. The gathering of feedback and insight from ISGs are central to this responsibility. Additionally, producing research is also important.

The second actor group is the ISGs, operating in countries such as India, Tanzania, Mozambique and Malawi. These are globally distributed groups all around the globe, tasked with implementing DHIS2 in their respective region's use context, and may also develop local apps, which often utilize DHIS2 functionality.

These groups are in a partnership with the DHIS2 core team but are independently run. Their projects typically entail creating solutions on contract for a project owner (often an NGO, e.g. UNICEF, NORAD, WHO), or the MoH in the respective country, or by innovating new apps locally. In addition, they are also often tasked with training users to the new solutions, and/or with maintenance of the solution. In addition to having contact with the DHIS2 core team in Oslo, the ISGs typically also have close ties with other ISGs. Figure 18 provides a model of the actors in the DHIS2 ecosystem.

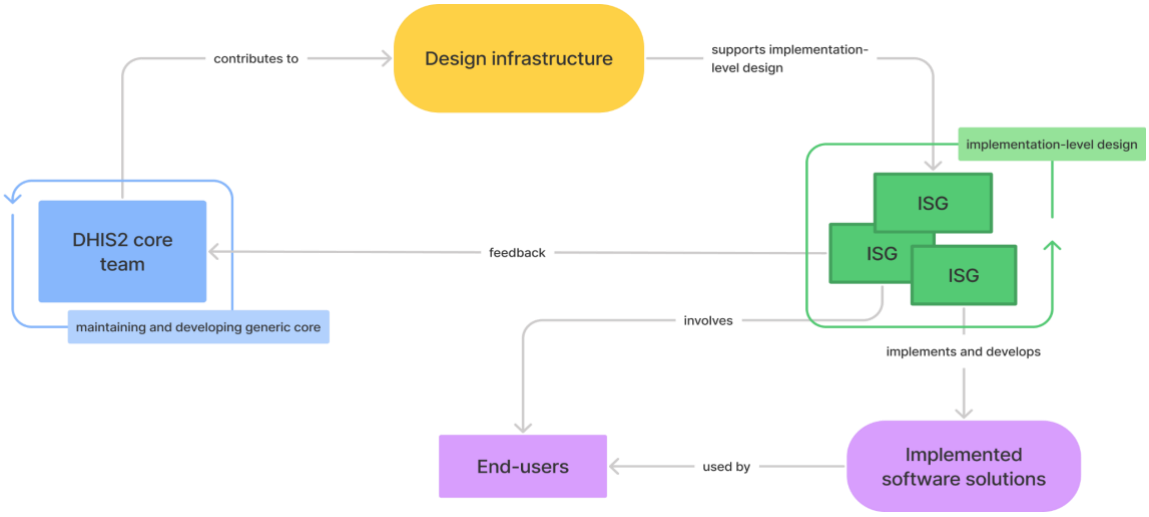

Figure 18: Overview of the DHIS2 ecosystem

Table 13 provides an overview of the DHIS2 ecosystem, with a short description of each part.

Part of ecosystem	**Description**
DHIS2 core team	● Vendor of DHIS2 ● Three main groups ○ DHIS2 product ○ DHIS2 coordination and management ○ Researchers ● Maintains and iterates on the design infrastructure ● Develops the generic core software with generic-level design
ISGs	● Independent implementation partners with HISP ● Uses implementation-level design to implement DHIS2 into local contexts ● May develop custom apps on top of DHIS2 ● Often does work on contract by an NGO or local Ministry of Health
End-users	● Intended users of the implemented software ● Will ideally participate as part of ensuring usability during implementation-level design ● Example: clinician workers, data entry clerks,
Design infrastructure	● Contains resources for DHIS2 ● Aids ISGs and DHIS2 core team in: ○ Implementation ○ Development ○ Training

Table 13: Overview of the actor groups within the DHIS2 ecosystem

5.2.2 What is the DHIS2 Design Lab?

Most of the projects in the Design Lab typically are focused on contributing to the DHIS2 core team, the design infrastructure, and/or one or more ISG. It consists of a multi-disciplinary group of master students, with backgrounds from development and/or user-centered design.

The Design Lab recently moved into new offices. The participants have access to meeting rooms, study rooms, and a large common lab-room. In addition, there are established direct messaging groups, as well as frequent meetings over Zoom due to restrictions set by the Covid-19 pandemic. The Design Lab has a firm focus on arranging its own events, such as seminars and social arrangements. Lab members are actively encouraged to give feedback on, and to ideate new events that may be useful. Lab members have also expressed that this engagement is an important part of being a participant. They want motivation and productive pressure, and to be part of a larger whole when working with the master projects. These ideas for events are also kept in the lab and are re-used and reiterated on several times.

These events are important, due to them facilitating co-learning between projects. Periodically, the lab arranges dedicated workshops (figure 19), where members can suggest new events or activities. Examples of such activities are giving constructive feedback on chapters, being participants in co-analysis sessions, or aid the progress of the research. Further, lab members have suggested activities such as extended abstract readings, peer-review sessions, or arranging seminars on research theory and methodology, as well as social events.

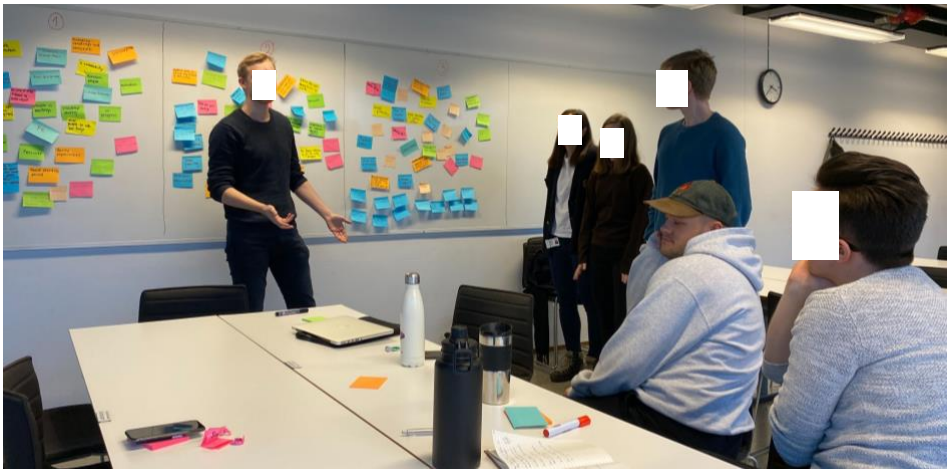

Figure 19: Seminar brainstorming on future lab events

As the lab grows in size and is starting to solidify as an entity, its presence becomes more known within the whole DHIS2 ecosystem. The lab acts as a physical as well as metaphorical meeting ground, where members from the DHIS2 core team and members from ISGs can meet. It also creates connections with the DHIS2 core team and ISGs due to the projects involving them. These connections are important to the DHIS2 core team: for instance, as told by a senior DHIS core team member: "*The lab should not govern local [ISGs], but rather be directly connected with the people. The lab is where the people are*". The field trips help establish these connections and are key to communicating with the ISGs. Further, by involving ISGs in the creation of the tools, the ISGs become more aware of the tools, and incentivise their utilization.

5.2.3 Design Lab contributions

In order to strengthen usability, the Design Lab contributes to three parts of the DHIS2 ecosystem. This section will describe these three and exemplify by using ongoing projects. Figure 20 gives an overview of the Design Labs position within the DHIS2 ecosystem, as well as how the lab makes its contributions to the different actors.

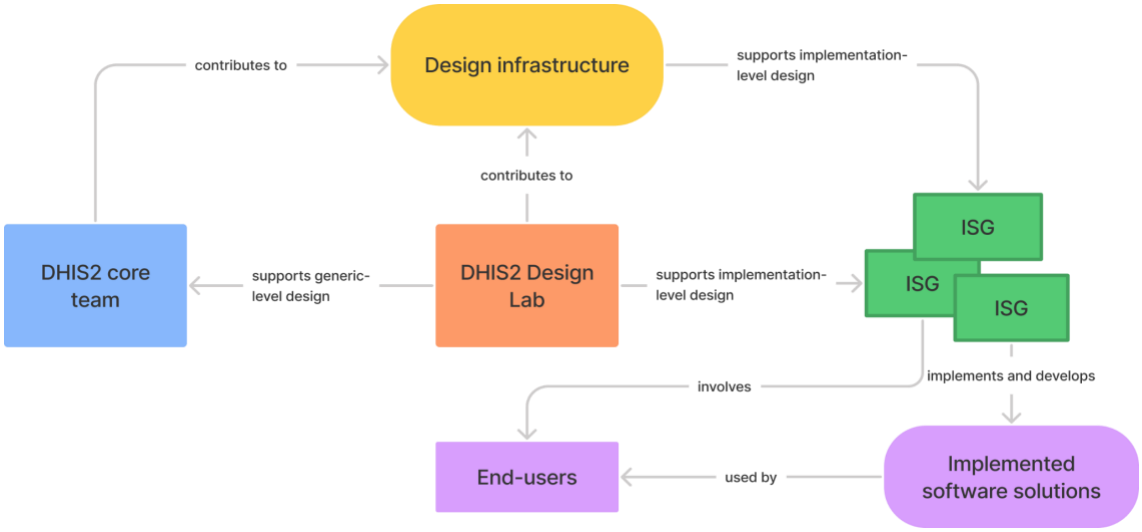

Figure 20: The DHIS2 Design Lab and how it contributes to the DHIS2 ecosystem

1: The lab contributes to the design infrastructure

The Design Lab contributes to the design infrastructure by creating resources aimed at strengthening implementation-level design. One of the currently active projects in the lab is concerned with the creation of a design method toolkit, containing resources providing guidance in involving users during implementation-level design. These resources are based on insights gathered from ISGs working with implementation-level design. By supporting the processes of involving end-users, the resource contributes to strengthening the usability of the final implemented solution.

Another currently active project aims to make app development less time and resource-intensive, allowing more time to be spent on usability-related activities. This is done through creating a component platform that facilitates component reuse, which makes developing custom apps a less time intensive activity. Another ongoing project with a similar aim is in the process of creating and iterating upon learning resources for the process of developing custom apps. By building development competence, it directly contributes to reducing the high requirements of time and competence, as argued in section 5.1.1. Ultimately, this leads to more usable solutions, due to the possibility of developing custom apps during implementation-level design.

2: The lab contributes to DHIS2 the core team

The DHIS2 Design Lab also contributes to the DHIS2 core team, by sharing research and insights concerning the processes of generic-level design. One ongoing lab project is concerned with how the DHIS2 core team can efficiently apply changes to the generic core software. Configurations done during implementation-level design are for instance in some cases relevant to apply to the generic core software (referred to as *generification* by for instance Monteiro et al. (2013)). By identifying emerging challenges during implementation-level design, and possible solutions during this process, the contribution facilitates usability during generic-level design in the DHIS2 core team. As argued in section 5.1.1, knowledge about constraints set by lacking flexibility when configuring DHIS2 can lead to usability issues. If these are communicated to the DHIS2 core team, they have the opportunity to remedy these challenges.

In addition, more general knowledge about the ISGs and their practices can also be useful to the DHIS2 core team. For instance, uncovering the steps may also prove useful to the implementation team. "*Having master students understand local practices is gold!*". Findings from field trips may prove valuable to the team and should be communicated.

3: The lab contributes to the ISGs

Finally, the lab also makes contributions to the ISGs. One project currently is collaborating in the development of a report generator with one ISG in Rwanda. By engaging in actual app development, lab members can contribute with their knowledge and expertise in a real project, based on local requirements gathered by ISG members. This is a remedy to the time-cost and required competence we argue is needed in section 5.1.1. By working directly with the ISG, the aforementioned general knowledge is also gathered, and is of interest to the DHIS2 core team.

To summarize: The Design Lab directly contributes to the design infrastructure, the DHIS2 core team, and the ISGs through the ongoing projects. Through these projects, general knowledge about DHIS2, the ISGs, and the processes of implementation and generic-level design are gained and should be shared with the DHIS2 core team.

5.3 Opportunities for strengthening the software usability

Until now, this chapter has looked at current challenges for usability within the DHIS2 ecosystem, and how the DHIS2 Design Lab relates to the ecosystem. This section describes which opportunities the DHIS2 Design Lab have to strengthen the software usability, based on our empirical insight of challenges, and the lab position within the ecosystem. These are opportunities either currently ongoing, or possible directions that may be taken in the future.

5.3.1 Internal Design Lab dynamics

By working towards keeping projects "alive" between projects, it can allow larger contributions to be made, and to iterate upon earlier work done. One approach to do this is to visualize the projects, which are done today by creating posters of ongoing and finished projects (see figure 21). Each project also has its own page on the DHIS2 website[1].

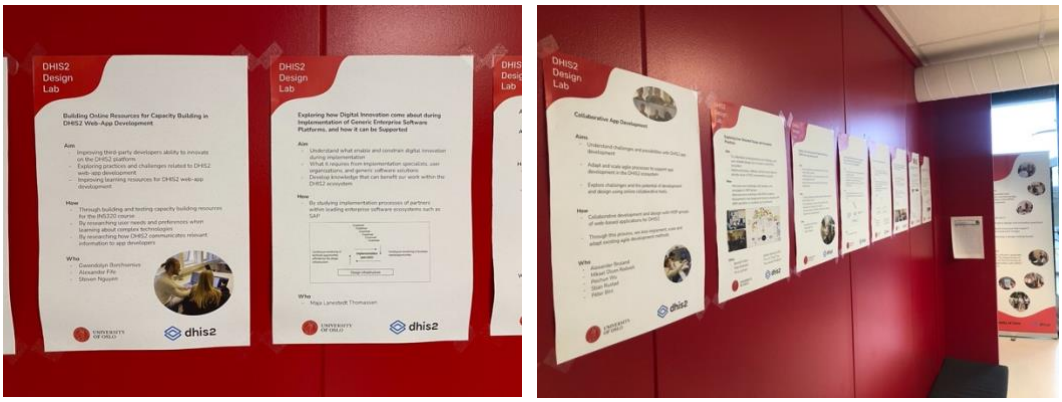

Figure 21: Project posters in the DHIS2 Design Lab office

Throughout two workshops with lab members, one of the emerging themes was that the lab should facilitate constructive communication between relevant members of the lab (see figure 22). Sharing insight from data collection activities may prove useful to other

[1] https://www.mn.uio.no/ifi/english/research/networks/hisp/dhis2-design-lab/

members, for instance eliminating the need to interview a certain person, freeing up time that may be spent elsewhere. The shared knowledge may also serve as complementary insight, providing a broader and deeper understanding. This collaboration is facilitated through regular seminars and meetings, where members present current learning goals for the project, as well as new findings. The lab provides academic support for the members. For instance, according to one student, the main reason to join the lab was "*to exchange knowledge*", and for the lab "*to teach me more about DSR [design science research], methodologies, and to know about [lab members'] research*".

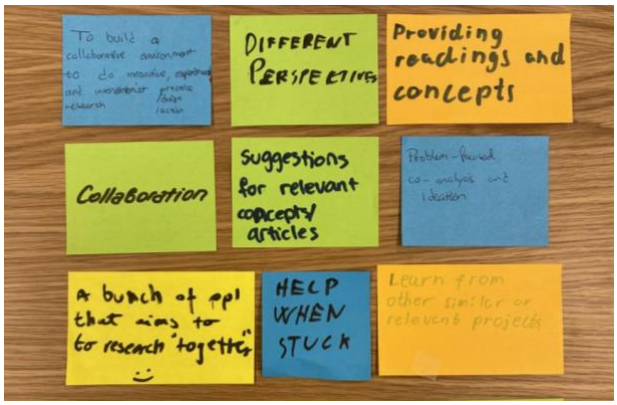

Figure 22: Post-it notes with suggestions written by lab members

These internal dynamics are important for the lab to function well in its goal of strengthening usability within the DHIS2 ecosystem. While not addressing usability directly, it helps the Design Lab members in their project work, which in turn can lead to more usable software in the end.

5.3.2 Innovating and iterating in the design infrastructure

As argued in section 5.1.1 and 5.1.2, creating resources in a design infrastructure aimed at supporting implementation-level design is central to achieving software usability. The Design Lab has ample opportunity to contribute to this infrastructure through the ongoing and future projects. Additionally, there is ample opportunity to innovate new resources in addition to the typical resources, such as documentation and design systems.

Especially training material and development tools to support development of custom apps have been mentioned as being in need for feedback. By being somewhat detached from the DHIS2 core team, the Design Lab members can innovate and ideate new concepts and ideas that members the DHIS2 core team may not be aware of.

For instance, we uncovered that prototyping is an important step in the development processes. Software such as Balsamiq are used to prototype data flow and the structure of the solution, while Adobe XD are used to map out more detailed user interfaces. When discussing these processes, several ISG members expressed an interest in getting more documentation and guidelines for these activities. The same interest was found when discussing user evaluation and testing as well.

5.3.3 Supporting generic-level design

In one of the workshops with the Design Lab members, some of the most voted-on ideas were for the lab to facilitate closer relationships with the DHIS2 core team, and to arrange seminars where they can present their work and what they do (see figure 23)[2]. By connecting the Design Lab and the DHIS2 core team, the Design Lab can contribute to generic-level design. For instance, projects such like the one mentioned in section 5.2.3 directly contributes to generic-level design.

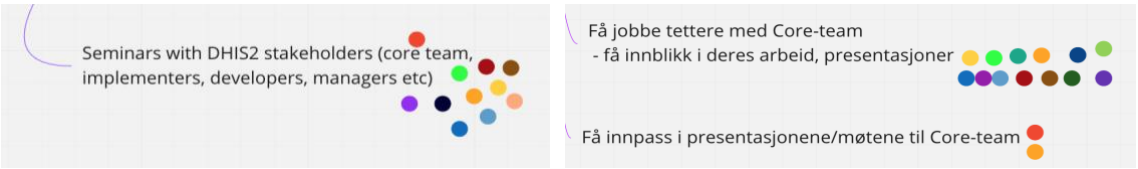

Figure 23: Votes (represented by a dot) on cooperating with the DHIS2 core team

The fact that the DHIS2 Design Lab today is somewhat disconnected with the rest of the core may also be seen as a positive factor. A DHIS2 core implementation coordinator told "*the lab can act as a breath of fresh air*" and that "*the members of the lab provide value*

[2] The workshop was arranged by us as part of the data collection in this thesis.

through being think-outside-the-box-people". This notion was also echoed by the DHIS2 core team members, working with generic-level design. Simply having more people working with generic-level design is useful, as there are limited resources allocated to focusing on generic-level design.

5.3.4 Connecting ISGs and the DHIS2 core team

The DHIS2 core team also expressed an interest in the lab creating connections and working directly with the ISGs. Before the restrictions due to the Covid-19 pandemic, there were plans to create a student exchange arrangement between the ISG in Mozambique and the lab, where both parts had students visiting each other over an extended period of time. While these plans currently are on hold, there is still an interest from the Design Lab members to go on field trips. As noted by one member: "*the reason for joining the lab was honestly the fact that I could travel*." and "*my original project involved me spending 4 months in Mozambique, where I would participate in a real project the [ISG] was working on*". This was also reinforced by findings from the workshops, where field trips were voted as the most important or relevant idea the Design Lab could arrange. Another finding of note was the interest in facilitating cooperation between ISGs and the Design Lab and its members.

Losing contact with the ISGs are a concern, from both the developers' and the implementers' perspective. As told by a front-end architect: "*the lab should work towards exposing the core members to the real world*". From a DHIS2 core team member tasked with coordinating implementation efforts: "*I need to understand what people are doing and what works and what does not work. If everyone moves a box as part of the implementation process, then maybe that box should be moved in the core software.*" By bringing back feedback from field trips and from working directly with the resources, new insights accumulate within the lab, and should be shared with the DHIS2 core team.

Additionally, the DHIS2 core team expresses a definitive interest in establishing channels between the lab and the DHIS2 core team, and the ISGs. While this connection to a certain degree already exists directly between the DHIS2 core team and the ISGs, the Design Lab may prove useful as a middleman or mediator between the two. It seems forging

connections directly with the ISGs, working together will be important to gaining and exchanging knowledge.

5.3.5 Using lab knowledge is difficult

As the lab is currently in the process of establishing itself as an institution, there are few solid connections to the DHIS2 core team where information can flow. Ensuring the flow of this type of knowledge is central to utilizing the potential of the Design Lab. As told by a central person in the DHIS2 core team: "*There is a lot of knowledge in the lab, but it stops there*", and *"It is a **** tragedy if master student work disappears. [..] We should break the habit of not using student knowledge".*

Typically, master projects follow a silo structure, where each project exists and is worked on separately from other projects. Traditionally, the project itself does not specifically facilitate, or invite cooperation or knowledge sharing between projects.

Further, as a result of the limited timeframe of a master project, the lab has an issue with creating continuity between master projects. Typically, after a project has concluded and the thesis is submitted, there is little incentive to keep the project alive, and continue the work previously done. As noted by the leader of the lab: "*The worst that can happen in an organization is that people quit often, every two years like master students do*".

5.3.6 Establishing local labs

One interesting finding was that the ISG in Mozambique also wanted to establish a local design lab, based upon the Design Lab in Oslo (figure 24).

Figure 24: Working in the newly established Design Lab in Mozambique

One of the senior members of the ISG said: "*We want to focus on how the lab is driven in Oslo, and also how a local lab can be driven and how they can collaborate in order to exchange knowledge and experience*". The goal of this local lab would be to 1) facilitate cooperation between the two labs, and 2) to serve as a local space for both ISG employees and students from the local university to innovate and cooperate with each other. The lab would become a competence center, meaning a space for looking at how design can be done, how methods are used, and to co-innovate. This would directly contribute to implementation-level design at the ISGs. Following the structure in the Design Lab, this local lab would also have a focus on including students from the local university in the development projects. Section 5.3 is summarized in table 14.

Opportunity	**Strengthens usability by**
Strengthening internal lab dynamics	Internal dynamics support the projects, leading to better contributions to generic- and implementation-level design
Innovating resources in the design infrastructure	Resources reduces time-cost, and facilitates competence used during implementation-level design
Supporting generic-level design	Implementation-level design hinges on generic-level design
Exchanging knowledge from ISGs to the DHIS2 core team	Generic-level design requires insights from the ISGs
Exchanging knowledge from the Design Lab to the DHIS2 core team	Learnings generated during projects in the Design Lab contributes to generic-level design
Establishing local Design Labs	Strengthening implementation-level design from a local process

Table 14: Summary of section opportunities

5.4 Chapter summary

Achieving usability is a joint effort between generic and implementation-level design. There are two approaches to doing implementation-level design. The first approach is to configure the generic software to fit the context of use, as well as end-users' needs. This process is relatively quick and easy, but there may exist severe limitations on what may be configured by the generic software. These limitations may be worked around or ignored, which often leads to a finished solution with usability issues. Additionally, the ISGs can submit a request for the configurability to be added, but this often takes too long to be a realistic fix. This approach sacrifices usability for being less time intensive, and requires less competence.

The second approach is to develop custom applications which are connected to the DHIS2 core software. This approach removes the constraints set by the configurability of the core software, allowing for the creation of a more usable solution. Developing these custom apps is a significantly longer process, and requires more competence. The result is that developing these apps are often not done due to the cost. By creating resources in a design infrastructure that support this process, reducing time and competence requirements, more custom apps may be developed, ultimately leading to a higher degree of usability. This task of creating resources is taken on by the projects in the Design Lab.

Like implementation-level design, generic-level design is strongly dependent on insight from end-users. As the end-users of the product of generic-level design are the ISGs, the DHIS2 core team are in great need of having insight in the processes of implementation-level design, and to understand any issues the ISGs have.

The Design Lab directly contributes to the design infrastructure, the DHIS2 core team, and the ISGs through the ongoing projects. Through these projects, general knowledge about DHIS2, the ISGs, and the processes of implementation and generic-level design are gained, and should be shared with the DHIS2 core team to strengthen the generic-level design.

Chapter 6 - Analysis and Discussion

To answer the research question of this thesis, we outlined four research objectives. This chapter addresses the final fourth: 4) Identify and discuss roles a design lab can play to strengthen the software usability in enterprise software ecosystems based on our empirical case. First, we discuss the DHIS2 Design Lab in relation to the existing literature on design labs and usability in enterprise software ecosystems, to conceptualize what a design lab is in the context of enterprise software ecosystems. Then, we present four roles a design lab can play to strengthen software usability derived from our empirical insight and will discuss these in relation to the literature presented in Chapter 3 to answer the research question of this thesis:

What roles can a design lab play to strengthen the software usability within enterprise software ecosystems?

Further, we discuss the limitations of our research, as well as implications for further research.

6.1 Conceptualizing the generic software design lab

Generic software is created through a two-step design process. Step one includes developing a generic core software, which in step two are implemented into a particular context of use (Bansler & Havn, 1994)**.** There are two processes of design; generic-level design where the vendor designs the generic core software, and implementation-level design where this core is implemented into a particular context of use by a implementation partner. Usability is a joint effort between the two processes, which if deficient, may lead to usability issues.(Sia & Soh, 2007; Soh et al., 2000; Strong & Volkoff, 2010)

A generic software design lab recognizes and attempts to support both processes of generic and implementation-level design. Recognizing that there are two processes of design are the main differences between a generic software design lab, and existing conceptualizations of design labs. The conceptualizations of the design collaboratorium,

the Design:Lab, and living labs all highlight involving actual end-users in the activities and processes of the design lab (Alavi et al., 2020; Binder & Brandt, 2008; Bødker & Buur, 2002), which are also supported by for instance UCD (Gulliksen et al., 2003; Norman, 2013) and PD (Simonsen & Robertson, 2012). We argue, however, that these traditional ways of involving users are unfeasible when done within the context of enterprise software ecosystems. Design within enterprise software ecosystems are not done at one place at one time, but rather simultaneously over longer periods, and are happening at multiple places, each with their local practices of doing implementation-level design (Dittrich, 2014). Rather than involving users directly, there is a need for supporting the processes of implementation-level design (Titlestad et al., 2009), which in turn has a focus on involving end-users. These processes of implementation-level design are also dependent on the design of the generic software, through the process of generic-level design done by the software vendor. Deficient generic-level design results in software that are too rigid, and not able to be configured, or allow development of custom apps during implementation-level design. Generic software usability depends on both generic- and implementation-level design, thus, the design lab is attempting to address both of these processes.

Table 15 illustrates the relationship between the design lab conceptualizations previously conceptualized and levels of design.

Design lab conceptualization	**Generic-level design**	**Implementation-level design**
Usability laboratorium		x
Design collaboratorium		x
Deisgn:Lab		x
Living lab		x
Generic software design lab	x	x

Table 15: Relationship between the design lab conceptualizations and levels of design

A generic software design lab addresses the actors in an enterprise software ecosystem. Bosch (2012) and Hanssen (2012) describe three key actor groups within enterprise software ecosystems: the vendor, implementation partners, and the end-users (Bosch, 2012; Hanssen, 2012). The generic software design lab considers all these actors as illustrated in figure 25. In the empirical case of this thesis, the DHIS2 core team acts as the vendor, the ISGs as the implementation partners, in addition to end-users. Based on this description, we argue the positioning of the DHIS2 Design Lab within the DHIS2 ecosystem is in accordance with the descriptions of actors in an enterprise software ecosystem. We argue these issues are closely linked to the processes of generic and implementation-level design, and argue these are the processes that have to be supported, in order to strengthen usability (figure 25). Thus, we also argue the processes of implementing DHIS2, and how this process can lead to usability issues is in accordance with the related body of literature.

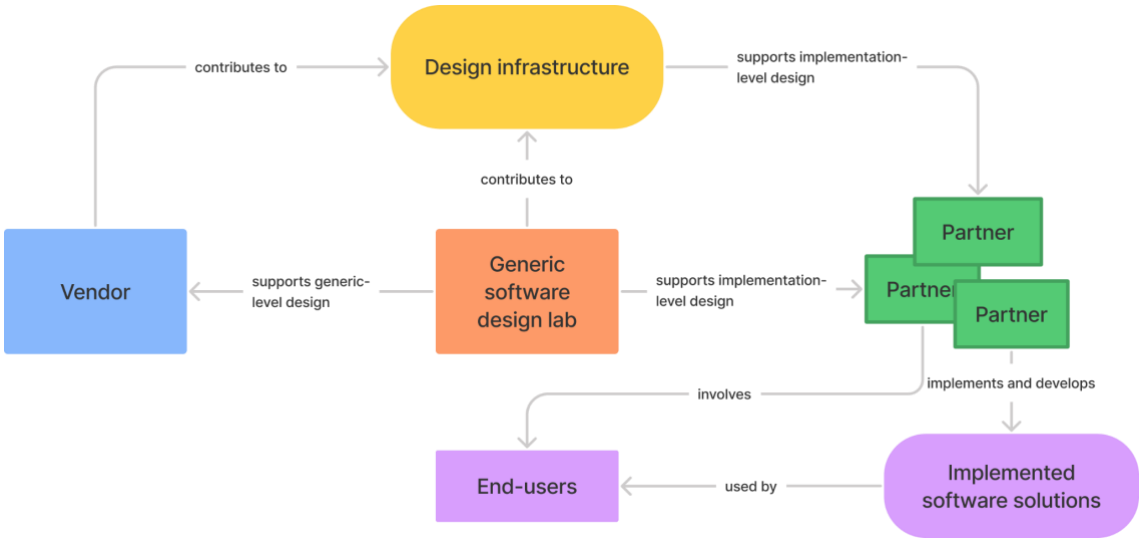

Figure 25: The generic software design lab in relation to the actors in an enterprise software ecosystem, and where it contributes to design

A generic software design lab offers a meeting ground where experiences, challenges and needs are shared between actors (for instance developers, members of the implementation team), and can potentially be addressed. This is a goal supported in both the conceptualization of the collaboratorium by Buur & Bødker (2000), as well as in the

Design:Lab by Binder & Brandt (2008). In both conceptualizations, it is argued a lab should act as a meeting ground, where a multidisciplinary group of people from both inside the lab as well as the outside can meet and create together.

A generic software design lab can ideate and create its own internal processes and activities, to stay relevant to its context. Since the lab is an evolving project, so are the processes within the lab. Such processes can for instance be events such as onboarding sessions, regular meetings, seminars on research topics, social gatherings, or standup-type regular check-ins. The creation of these events is important for the internal workings in the lab: the processes should support the work done, in order to help the contributions aimed at supporting generic and implementation-level design. There is both a focus on innovating new processes, as well as iterating and changing existing processes. Due to the lab being quite self-governed and having a flat hierarchical structure, there is great opportunity to ideate and arrange new events based on ideas from either the leader, or the students in the lab. The argument has been made that it is important to be able to innovate and create relevant processes by utilizing the generative space the lab is (Binder & Brandt, 2008). These processes are then stored in the lab and can be reused several times.

A generic software design lab stores and iterates on previous work. Binder & Brandt (2008) also argue the lab can store and/or display artefacts resulting from the work done. Bødker & Buur (2002) also argue that the collaboratorium should both accumulate useful insights informing design, as well as inspire innovation. The design lab acts as a record keeper of earlier work. The resources created in the lab serve as artefacts stored in the lab. One example of an artefact from the DHIS2 Design Lab is the design method toolkit. Other artefacts such as project posters have been created, and may be expanded by for instance photos, models, and physical prototypes. These records may then facilitate further ideation. Using record keeping in traditional laboratories as a metaphor, Binder and Brandt emphasizes the importance of reiterating on previous learnings:

"[..] *We need means of documentation that act like records of the experiments to maintain, accumulate and continuously reiterate what is learned. The laboratory metaphor can help*

to ensure that we do not end up with collaborative events that are fun in themselves but do not leave a lasting imprint on the inquiry" (Binder & Brandt, 2008, p. 120).

6.2 Roles

In the following section, we present the roles we have identified. Table 16 gives a summary of the roles a design lab can play to strengthen software usability in enterprise software ecosystems, and the roles are elaborated and discussed below.

Role	**Description**
Engaging with implementation partners during implementation-level design	Contributing resources supporting implementation-level design, by for instance building implementation partner competence. Participate and contributes to projects concerning implementation-level design
Engaging with vendor during generic-level design	Participate and contributes to projects concerning generic-level design Report experiences from implementation-level design to vendor
Design champion	Promote the value of working with design to address software usability Makes design more relevant by drawing more attention towards design work - putting design on the agenda of implementation partners and vendors. Inspiration for local labs
Workspace	Facilitates knowledge sharing between projects and ensures finished projects can be worked on further. Serves as a meeting ground for internal members, and external guests

	Ideates practices and activities relevant to the context of the generic software design lab Increases the contributions made to design infrastructure, promoting usability during implementation-level design

Table 16: Overview and short description of identified roles

Engaging with implementation partners during implementation-level design

Our empirical findings show that a design lab can play a role in strengthening the software usability through contributing resources to a design infrastructure. These contributions consider the implementation partners and their needs in the implementation processes (Dittrich, 2014), by providing artefacts specifically created for implementation-level design (Titlestad et al., 2009)**.** Baxter and Sommerville (2011), for instance, specifically calls for process guidance for using design methods, which the lab design lab can contribute to this infrastructure.

Based upon the challenges and opportunities outlined in our findings, we identify several examples of resources that are valuable to implementation partners during implementation-level design. Examples are: training material, application development resources, and resources for user prototyping, contract negotiation, testing and evaluations. In addition to directly supporting the processes of implementation-level design, it allows the design lab to provide a novel perspective on the creation of these resources, complementing the viewpoints of the vendor. In the DHIS2 core team's own words; by being a "breath of fresh air". Additionally, the design lab can actively participate in implementation-level design. Usability is strengthened by contributing with competence from the lab members, as well as "man-power", taking part in tasks of implementing the software.

Contributing to the design infrastructure also provides a way for a design lab to stay relevant in an enterprise software ecosystem. The fact that the design lab addresses usability through the infrastructure rather than directly, it considers the implications from the concept of meta-design by Ehn (2008), and the division of design for use, and design for design by Fischer & Giaccardi (2006). This makes it possible to address usability with

consideration to the division between generic and implementation-level design. We argue the traditional approaches of designing for end-users from the perspective of the vendor creates a conflict in how to 1) include a significant variety of users from different use-cases, and 2) combine the insight to create a "one-size fits all"-solution. However, by addressing usability through the design infrastructure, it can create resources that support implementation-level design in addressing usability

Engaging with vendor during generic-level design

Efforts to support generic-level design are needed to facilitate further design on the level of implementation. By engaging with the vendor during generic-level design and providing support in this process, a design lab can play a role in strengthening the software usability. This can for instance involve reporting challenges in addressing usability directly from implementation-level design to the vendor, or develop learnings on how vendor efficiently can apply changes to the generic core software based on feedback from implementation partners.

Our empirical findings show that the DHIS2 core team aims to provide generic functionalities to the generic core software that are useful in a variety of different contexts. As a part of this work, the results derived from projects of implementing DHIS2 are ideally fed back to the DHIS2 core team so they can consider adding that functionality to the generic core software. If the functionality is considered valuable to several ISGs, it will typically be included. With this strategy, continuous contact with the ISGs is crucial. However, both the DHIS2 core team and the ISGs expressed that keeping connected with each other is a challenge. The lack of communication slows down the desired evolution of the generic core software, which consequently leads to less relevant functionality to support implementation-level design.

Another consequence from the lack of communication between the DHIS2 core team and ISG is the shortage of information on challenges the ISGs have during implementation-level design in regards to configurability and custom app development. This results in unfulfilled needs for ISGs and the end-users. What ISGs and end-users need evolves with time, which makes staying in contact also a measure of understanding the evolving

contexts of use. The DHIS2 Design Lab currently explores ways communication channels between the DHIS2 core team and ISGs can be optimized. Learnings from this project inform how desired changes in the generic core software are noticed and acted upon, resulting in a more informed work practice for the DHIS2 core team

Dittrich (2014) reports on communication issues in four enterprise software ecosystems, and highlights the importance for communication to flow between the actors involved, especially in regards to accomplishing continuous evolution of the generic core software. Additionally, keeping connected with the context of use is argued to be important by Dittrich (2014), as it is the main source of requirements and inspiration for generic-level design. If the generic software is continuously updated with relevant functionality, design on the level of implementation can result in more usable and relevant outcomes for the end-users.

As seen in our empirical findings, field trips are encouraged as part of the Design Lab members research work, and on these trips valuable knowledge on project results and challenges such as limited configurability or user interface (UI) problems, experienced during implementation-level design are gained. The Design Lab members usually arrange for presentations to share interesting insights with each other after every field trip. This is a convenient arena for the DHIS2 core team to participate. Exposing the DHIS2 core team to what happens in "the real world" can help with the evolution of the generic core software, as well as improve the current functionality. The field trip presentations show to be of great interest to the DHIS2 core team. However, due to the little time they have available, measures for how the insights can be communicated efficiently without them having to spend hours listening in on presentations is useful to explore further. Regardless of how the insights are communicated, the DHIS2 core team expressed a keen interest in utilizing the knowledge gained by the DHIS2 Design Lab.

By engaging in generic-level design and supporting this process, a design lab can report challenges in addressing usability directly from implementation-level design, and work to find solutions to evolve the generic productively to fit the contexts of use.

Communication between vendor and implementation partner is then also leveraged which is argued to be an important element in a enterprise software ecosystem (Dittrich, 2014).

Design champion

The design lab also plays a role as a design champion within the enterprise software ecosystem. By working with the implementation partners and the vendor, design as a topic is put on the agenda of both actors. Through our empirical findings, we saw that there are few people within the DHIS2 core team that are working only with design, and that there was a concern regarding the lack of focus on design within the DHIS2 core team as a whole. An extensive body of literature argues having a focus on use-oriented design (Ehn, 2008; Kujala, 2003; Li, 2019a; Norman, 2013; Rosson & Carroll, 2007). The existing conceptualizations of design labs also all have a common focus on **the user**, in an effort of creating a usable solution (Alavi et al., 2020; Binder & Brandt, 2008; Buur & Bødker, 2000; Nielsen, 1994). Yet, usability issues are still a challenge within generic software (Sia & Soh, 2007; Soh et al., 2000; Strong & Volkoff, 2010). Thus, the design lab contributes to pulling a much needed focus towards design and usability. The ISGs also expressed interest in utilizing methods and processes from Oslo, further strengthening the labs position as "the place" concerned with design.

An interesting aspect of the role as a design champion is to establish several design labs within a single enterprise software ecosystem. Taking advantage of the different locations of the implementation partners, having several design labs spread across the globe would provide a broader perspective on how implementation-level design is done, and of the challenges the implementation partners face. It also opens the possibility of having several labs contribute resources for the design infrastructure. This would shift the development more towards a bottom-up perspective, both contributing through the time and resources spent on the resources, but also get complementary perspectives from several use contexts, and the issues and challenges they impose on implementation-level design. Secondly, it would also form a basis for creating a global design community, where lines of communication between the different design labs would allow better discussions on the topic of design.

Workspace

The DHIS2 Design Lab strengthens usability within DHIS2 by playing a role as a workspace for lab members. The focus on encouraging sharing of insights across projects can provide breadth and depth for the projects. For instance, sharing general interview insights from interviews from a particular person may provide complementary insight for another project, providing a broader view of the issue at hand. Despite not necessarily sharing a common room, this peripheral awareness of other members' work, we see similarities to the design collaboratorium. Buur and Bødker (2000) argue the design collaboratorium has different people working in close proximity, and that this peripheral awareness of each other's work may lead to direct collaboration.

Further, the design lab encourages using previous knowledge, iterating on already gathered insight. As a result, larger and more complex or contributions of greater value. As noted in Chapter 5, having people leave a team in a professional setting can be detrimental to the results produced. The limited time frame of the master projects set restrictions on the extent of the contribution that can be made. However, by ensuring project work is continued and is used in further projects, larger contributions to the design infrastructure can be made, in turn strengthening usability during implementation-level design. In short: making the design lab do its work better, the contributions it makes are also of better quality, resulting in better strengthening of usability.

6.3 Contribution

The contributions of this thesis are both practical and theoretical. We relate the theoretical contributions to two streams of research. The first stream is concerned with existing conceptualizations of design labs. We argue the existing approaches of the usability laboratory (Nielsen, 1994), design collaboratorium (Buur & Bødker, 2000), Design:Lab (Binder & Brandt, 2008), and Living Lab (Alavi et al., 2020) are ill-equipped in contexts where both generic and implementation-level design has to be considered. Existing lab approaches assume an aim of addressing usability at one time and place, which is not possible within enterprise software ecosystems (Monteiro et al., 2013). Additionally, these approaches do not recognize the implications of involving users when design is

done at two levels, through generic and implementation-level design. We make a theoretical contribution by conceptualizing a generic software design lab, and by outlining four roles a design lab can play to strengthen the software usability within enterprise software ecosystems. We view these roles as an extension to the already established lab approaches. Additionally, we argue the roles make a practical contribution by serving as a starting point for new generic software design labs, or serving as a fulcrum for taking existing labs in a direction closer to enterprise software ecosystems.

The second stream is concerned with addressing software usability within enterprise software ecosystems. Usability-related issues in the software (Sia & Soh, 2007; Soh et al., 2000; Strong & Volkoff, 2010) and approaches of addressing these issues are well-documented (Baxter & Sommerville, 2011; Dittrich, 2014; Li, 2019b) in existing IS literature. Our empirical data support the existence of these issues as we recognized a number of similar issues between enterprise software ecosystems in the literature, and within the DHIS2 ecosystem. Our second practical contribution provides insight into one approach for addressing these issues, and may serve as inspiration for new means for reaching the end of increased software usability. We also make a practical contribution for practitioners working with usability within enterprise software ecosystems. As a consequence of using an engaged scholarship-approach, the contribution may also be of practical importance to the DHIS2 Design Lab, the ISGs, and the DHIS2 core team, but also for other enterprise software ecosystems such as SAP, Salesforce and Microsoft Dynamics.

Table 17 summarizes our contribution in relation to usability issues in DHIS2, as well as problems discussed in the related body of literature.

Theoretical and practical problem		**Leading to usability problem**	**Addressed by role**
Problem described in existing academic literature	Communication and collaboration with implementation partners are a central issue, and are central to the continuous evolution of the software (Dittrich, 2014)	Implementation partners need resources aimed at strengthening implementation-level design Communication is key to understanding which resources are needed, and which challenges implementation partners face during implementation-level design.	Engaging with vendor during generic-level design Engaging with implementation partners during implementation-level design
DHIS2 Problem	The DHIS2 core team are losing contact with the ISGs Design obstacles due to limited flexibility during configuration is not communicated ISGs require resources during custom app development and configuration, but the DHIS2 core team do not know which resources are needed by the ISGs during implementation-level design		
Problem described in existing academic literature	Implementation requires a thorough and time-consuming process, and are not straightforward (Dittrich, 2014; Sia & Soh, 2007)	User involvement during Implementation-level design is time/cost intensive, and is deprioritized/not done at all due to time and resource cost.	Engaging with vendor during generic-level design Engaging with implementation partners during implementation-level design
DHIS2 Problem	Implementing and developing with DHIS2 is a time- and resource-intensive process. Design-focused efforts are being limited by contracts and mandates.		

Problem described in existing academic literature	Design is not a local activity, but spread across time and geographic places (Monteiro et al., 2013). Existing labs (Alavi et al., 2020; Binder & Brandt, 2008; Buur & Bødker, 2000; Nielsen, 1994) are not equipped to address usability at the generic- and implementation-level	Using existing lab approaches as a means to work with end-users during generic-level design cannot realistically be done	Workspace Engaging with implementation partners during implementation-level design
DHIS2 Problem	DHIS2 designers cannot feasibly include a significant amount of end-users at generic-level design		
Problem described in existing academic literature	Design labs has to prototype their own workings and processes (Bødker & Buur, 2002)	Internal lab processes are central for the lab to function in the other roles. If the processes are better, the results from the design lab are also better	Workspace
DHIS2 Problem	The lab has existed for a limited time, and has thus not developed a significant collection of processes. The lab has few formal connections with the DHIS2 core team		
Problem described in existing academic literature	Usability issues are a challenge within generic software (Sia & Soh, 2007; Soh et al., 2000; Strong & Volkoff, 2010), despite usability being widely accepted as important in software development	There is potential to address usability during both implementation and generic-level design through a design lab	Engaging with implementation partners during implementation-level design Design champion
DHIS2 Problem	There is a current lack of focus on design, limited personnel and time to work with design		

Table 17: Roles in relation to usability issues from literature and the empirical findings

6.4 Reflections upon the research conducted

As with all studies, our research is subject to limitations. We have in our research studied the DHIS2 Design Lab and how it can remedy the usability challenges experienced by two central stakeholders in the DHIS2 ecosystem: ISGs and the DHIS2 core team. We have conceptualized a generic software design lab and identified and discussed four roles a design lab can play to strengthen the software usability in enterprise software ecosystems and subsequently. In this section we will present the limitations that possibly have affected our contribution and provide directions for further research.

6.4.1 Limitations

In this section we will describe the limitations of our research due to the unforeseen Covid-19 Pandemic, and limitations of the methods we applied in our study.

The Covid-19 Pandemic

Soon after the return from our field trip to Malawi and Mozambique, the first cases of Covid-19 was reported in Norway and several other countries which shortly led to a global pandemic. The pandemic has resulted in significant disruptions in everyone's lives including closed offices, closed education institutions, lockdowns, and travel restrictions. This extraordinary situation had a notable impact on the research work we present in this thesis. Of significant relevance was our plan to travel back to Malawi and Mozambique and do further research on the ISGs work practice which was cancelled. The plans included close collaboration on exploring how the Design Lab in Oslo could connect to their ongoing efforts in strengthening usability-oriented design. Consequently we had to adjust the direction of our research and work out new plans to conduct an interesting and relevant study. Additionally, the activities in the DHIS2 Design Lab had to be moved online or be cancelled as a result of UiO being closed, which was a barrier for our lab participation and coordination data collection activity. All interviews and some of our data analysis had to be done digitally which we experienced as a challenging aspect of our research process. Of notable importance is also how the pandemic affected our capacity to write our first academic publication. Doing so is no unchallenging task, and is arguably increasingly challenging during a pandemic. Deficient work environments

and less of an opportunity to discuss and ideate with fellow researchers made the process significantly more demanding.

Methods

As interpretive researchers exploring a social phenomenon such as a "design lab", we brought with us our subjectivity in the process of understanding and making sense out of our data. Thus, the documented findings are shaped by our subjectivity. However, the collaboration and discussions with Design Lab members we managed to organize, and our interactions with a number of DHIS2 core team members, as well ISG representatives has been valuable to enrich our understanding. Being two researchers working together throughout the whole research process and in the writing of this thesis has also ensured a more nuanced picture.

Our study has involved the conduction of several methods for data collection aimed to answer our research objectives and research question. This has included an examination of the three stakeholders: ISGs, the DHIS2 core team and Design Lab members. We emphasize the opportunity to gain a deeper understanding of our research topic by conducting additional methods with the DHIS2 core team. By using multiple methods we could develop a more comprehensive understanding of the challenges they experience when addressing usability, and uncover aspects of this work that were potentially lost during the interviews.

The ISGs we visited in Malawi and Mozambique represent only a couple of the organizations implementing and configuring DHIS2, and so by interacting and collecting data with more ISGs we could be provided with a deeper understanding of our research topic. Moreover, it would be interesting to interact with the ISGs throughout the whole research process to uncover any further issues with their work of addressing usability.

We also emphasize that the roles we have identified in our research are not further discussed with the stakeholder we have based our empirical insights on. Conducting evaluation sessions where the ISGs, the DHIS2 core team and Design Lab members could

offer their opinion on if the roles we have identified is of value to strengthen software usability in enterprise software ecosystems, could give us more relevant research results.

6.4.2 Further research

The contributions presented in this thesis lay the foundation for interesting further research. We will here propose three directions for further research.

Due to the limited time we had to conduct our research, we would first like to address the opportunity to conduct a longitudinal study where the DHIS2 Design Lab is explored over a longer period of time. This approach is argued to be essential for studying organizational processes in IS research, and is especially valuable to identify cause and effect relationships (Street & Ward, 2012). Through a longitudinal study, emerging and evolving trends and qualities can be detected to a more significant degree than we were in the position to identify in our one-and-a-half-year-long research project.

Secondly, this is a single case study which affects the generalizability of our findings. However, we wish to highlight that the value of the roles we have identified is to serve as a starting point for new generic software design labs and how they can play a role in addressing usability. Further research is needed to address the complexity of enterprise software ecosystems and the work of addressing software usability during generic and implementation-level design. Design Labs and enterprise software ecosystems are evolving phenomena which cannot be viewed in isolation to its environment. The findings from our research are naturally shaped by HISP's and the DHIS2 Design Lab's organizational characteristics, such as the focus on academic research. We have little knowledge as to what degree academic research is utilized in other enterprise software ecosystems. For instance, the fact that the Design Lab members in our case are master students who are typically driven by other motivational factors than paid employees, who are compensated monetarily for their time and work, might yield other empirical findings in different cases.

Yin (2003) presents several arguments on why a single case study is favourable, including that the case is unique and that it can act as a preliminary effort to further research.

Building on this argument, we propose the need for further research on other enterprise software ecosystems to potentially uncover additional or other roles a design lab can play to strengthen the software usability.

Lastly, in this research we have identified a gap in the design lab literature on using lab-type approaches to address usability in the context of enterprise software ecosystems. We need to better understand the traits of generic software design labs and how they can strengthen software usability. We have through this research addressed this topic, but we see a need to investigate it further.

Chapter 7 - Conclusion

This thesis has examined what roles a design lab can play to strengthen the software usability within enterprise software ecosystems. Having explored the challenges a vendor and implementation partners face when addressing usability, we identified potential ways a design lab can remedy these challenges. Making enterprise software that is perceived as usable is well documented in academic literature to be a challenge (Martin et al., 2007; Sia & Soh, 2007; Strong & Volkoff, 2010), and is argued to emerge from the software's inherent characteristic to fit general use. Usability is an important feature in any software, enabling the users to achieve a desired efficiency, effectiveness and user satisfaction (Grudin, 1991; ISO, 2018). Traditional methods to address software usability emphasizes the importance of use-oriented design, considering end-users needs and practices (Ehn, 2008; Kujala, 2003; Li, 2019b; Norman, 2013; Rosson & Carroll, 2007). The established practice of designing for usability is, however, incompatible when it meets the process of developing generic software where design is not constrained to one time and place.

As an effort to address the problem of making enterprise software that is perceived as usable, we have conceptualized a generic software design lab, as well as identified four roles a design lab can play to strengthen the software usability in enterprise software ecosystems. The empirical foundation these roles are based on comes from a-one-and-a-half-year embedded case study where we followed the DHIS2 Design Lab, which attempts to address usability-related challenges within the DHIS2 ecosystem.

The conceptualization and the four roles provide insight as to how usability can be addressed by supporting the design of the generic core software, as well as strengthening the processes of implementing the generic core software in a particular use context. Based on our empirical case, we contribute to literature on enterprise software ecosystems and design labs by conceptualizing a generic software design lab, which takes into account the scale and diverse contexts of use of generic software. We further contribute by identifying four roles a design lab can play to address usability-related problems in generic software ecosystems. In addition to being relevant to

researchers, our conceptualizations and findings are relevant to practitioners concerned with design in enterprise software ecosystems.

References

Alavi, H. S., Lalanne, D., & Rogers, Y. (2020). The five strands of living lab. *ACM Transactions on Computer-Human Interaction*, *27*(2). Scopus. https://doi.org/10.1145/3380958

Andersen, T. O., Kanstrup, A. M., & Yndigegn, S. L. (2018). Three living labs in Denmark: Challenges with co-design and implementation of health IT. *Proceedings from The 16th Scandinavian Conference on Health Informatics 2018 Aalborg, Denmark August 28–29, 2018*, 1–6.

Bansler, J., & Havn, E. (1994). *Information systems development with generic systems.* 707–718.

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

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, *11*(3), 369–386. https://doi.org/10.2307/248684

Berente, N., Lyytinen, K., Yoo, Y., & King, J. L. (2016). Routines as Shock Absorbers During Organizational Transformation: Integration, Control, and NASA's Enterprise Information System. *Organization Science*, *27*(3), 551–572. https://doi.org/10.1287/orsc.2016.1046

Binder, T., & Brandt, E. (2008). The Design:Lab as platform in participatory design research. *CoDesign*, *4*(2), 115–129. https://doi.org/10.1080/15710880802117113

Bødker, S., & Buur, J. (2002). The design collaboratorium: A place for usability design. *ACM Transactions on Computer-Human Interaction*, *9*(2), 152–169. https://doi.org/10.1145/513665.513670

Bødker, S., Grφnbæk, K., & Kyng, M. (1995). Cooperative Design: Techniques and Experiences From the Scandinavian Scene. In R. M. Baecker, J. Grudin, W. A. S. Buxton, & S. Greenberg (Eds.), *Readings in Human–Computer Interaction* (pp. 215–224). Morgan Kaufmann. https://doi.org/10.1016/B978-0-08-051574-8.50025-X

Bosch, J. (2012). Software ecosystems: Taking software development beyond the boundaries of the organization. *Journal of Systems and Software*, *85*(7), 1453–1454. https://doi.org/10.1016/j.jss.2012.03.039

Braa, J., Hanseth, O., Heywood, A., Mohammed, W., & Shaw, V. (2007). Developing Health Information Systems in Developing Countries: The Flexible Standards Strategy. *MIS Quarterly*, *31*(2), 381–402. https://doi.org/10.2307/25148796

Braa, J., & Sahay, S. (n.d.). *The Process of Developing the DHIS*. Retrieved 4 May 2021, from https://www.mn.uio.no/ifi/english/research/networks/hisp/about/hisp-history.html

Braa, J., & Sahay, S. (2012). *Participatory design within the HISP network: Looking back and looking forward* (p. p 235-256).

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Buur, J., & Bødker, S. (2000). From usability lab to 'design collaboratorium': Reframing usability practice. *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, 297–307. https://doi.org/10.1145/347642.347768

Callén, B., Domènech, M., López, D., & Tirado, F. (2009). Telecare research: (Cosmo)politicizing methodology. *Alter*, *3*(2), 110–122. https://doi.org/10.1016/j.alter.2009.02.001

Chua, W. F. (1986). Radical Developments in Accounting Thought. *The Accounting Review*, *61*(4), 601–632.

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

DHIS2. (n.d.-a). *About DHIS2*. About DHIS2. Retrieved 4 May 2021, from https://dhis2.org/about/

DHIS2. (n.d.-b). *DHIS2 In Action*. DHIS2. Retrieved 15 May 2021, from https://dhis2.org/in-action/

DHIS2. (n.d.-c). *Information Systems Research and DHIS2*. Information Systems Research and DHIS2. Retrieved 25 March 2021, from https://dhis2.org/research/

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

Dittrich, Y., Vaucouleur, S., & Giff, S. (2009). ERP Customization as Software Engineering: Knowledge Sharing and Cooperation. *IEEE Software*, *26*(6), 41–47. https://doi.org/10.1109/MS.2009.173

Edwards, R., & Holland, J. (2013). *What is qualitative interviewing?* A&C Black.

Ehn, P. (2008). *Participation in Design Things*. 92–101. http://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-11060

Fischer, G., & Giaccardi, E. (2006). Meta-design: A Framework for the Future of End-User Development. In H. Lieberman, F. Paternò, & V. Wulf (Eds.), *End User Development* (pp. 427–457). Springer Netherlands. https://doi.org/10.1007/1-4020-5386-X_19

Geertz, C., & Books, B. (1973). *The Interpretation Of Cultures*. Basic Books.

Grudin, J. (1991). Interactive systems: Bridging the gaps between developers and users. *Computer*, *24*(4), 59–69. https://doi.org/10.1109/2.76263

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

Hanssen, G. K. (2012). A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. *Journal of Systems and Software*, *85*(7), 1455–1466. https://doi.org/10.1016/j.jss.2011.04.020

Hyysalo, S., & Hakkarainen, L. (2014). What difference does a living lab make? Comparing two health technology innovation projects. *CoDesign*, *10*(3–4), 191–208. https://doi.org/10.1080/15710882.2014.983936

ISO. (2018). *Ergonomics of human-system interaction—Part 11: Usability: Definitions and concepts*. https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en

Kanstrup, A. M. (2017). Living in the lab: An analysis of the work in eight living laboratories set up in care homes for technology innovation. *CoDesign*, *13*(1), 49–64. https://doi.org/10.1080/15710882.2016.1146304

Kjeldskov, J., & Skov, M. B. (2014). Was it worth the hassle? Ten years of mobile HCI research discussions on lab and field evaluations. *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, 43–52. https://doi.org/10.1145/2628363.2628398

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. 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. (2021). *Generic Enterprise Software implementation as Context for User-Oriented Design: Three Conditions and their Implications for Vendors.* 12th Scandinavian Conference on Information Systems (SCIS), Orkanger, Norway.

Li, M. (2019a). *Addressing usability in generic software – The holistic usability design lab*. 14.

Li, M. (2019b). An Approach to Addressing the Usability and Local Relevance of Generic Enterprise Software. *Selected Papers of the IRIS*, *10*, 1–15. https://aisel.aisnet.org/iris2019/3

Li, M., & Nielsen, P. (2019a). *Design Infrastructures in Global Software Platform Ecosystems*. 7.

Li, M., & Nielsen, P. (2019b). Making Usable Generic Software. A Matter of Global or Local Design? *Scandinavian Conference on Information Systems*. 10th Scandinavian Conference on Information Systems, Nokia, Finland.

https://www.researchgate.net/publication/332848767_Making_Usable_Generic_Software_A_Matter_of_Global_or_Local_Design

Martin, D., Mariani, J., & Rouncefield, M. (2007). Managing integration work in an NHS electronic patient record (EPR) project. *Health Informatics Journal*, *13*(1), 47–56. https://doi.org/10.1177/1460458207073648

Martin, D., Rouncefield, M., O'Neill, J., Hartswood, M., & Randall, D. (2005). Timing in the art of integration: 'that's how the bastille got stormed'. *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work*, 313–322. https://doi.org/10.1145/1099203.1099256

Mathiassen, L. (2002). Collaborative practice research. *Information Technology & People*, *15*(4), 321–345. https://doi.org/10.1108/09593840210453115

Mathiassen, L. (2017). Designing Engaged Scholarship: From Real-World Problems to Research Publications. *Engaged Management ReView*, *1*(1). https://doi.org/10.28953/2375-8643.1000

Mathiassen, L., & Nielsen, P. A. (2008). *Engaged Scholarship in IS Research*. *20*, 19.

Monteiro, E., Pollock, N., Hanseth, O., & Williams, R. (2013). From Artefacts to Infrastructures. *Computer Supported Cooperative Work (CSCW)*, *22*(4), 575–607. https://doi.org/10.1007/s10606-012-9167-1

Mousavidin, E., & Silva, L. (2017). Theorizing the configuration of modifiable off-the-shelf software. *Information Technology & People*, *30*(4), 887–909. http://dx.doi.org.ezproxy.uio.no/10.1108/ITP-07-2014-0140

Nielsen, J. (1994). Usability laboratories. *Behaviour & Information Technology*, *13*(1–2), 3–8. https://doi.org/10.1080/01449299408914577

Norman, D. A. (1998). *The invisible computer: Why good products can fail, the personal computer is so complex, and information appliances are the solution*. MIT press. https://mitpress.mit.edu/books/invisible-computer

Norman, D. A. (2013). *The design of everyday things* (Revised and expanded edition). Basic Books.

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research, 2*(1), 1–28. https://doi.org/10.1287/isre.2.1.1

Pollock, N., Williams, R., & D'Adderio, L. (2007). Global Software and its Provenance: Generification Work in the Production of Organizational Software Packages. *Social Studies of Science*, *37*(2), 254–280. https://doi.org/10.1177/0306312706066022

Rosson, M. B., & Carroll, J. M. (2007). Scenario-based design. In *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 1032–1050). L. Erlbaum Associates Inc.

Sæbø, J. I., Kossi, E. K., Titlestad, O. H., Tohouri, R. R., & Braa, J. (2011). Comparing strategies to integrate health information systems following a data warehouse approach in four countries. *Information Technology for Development*, *17*(1), 42–60. https://doi.org/10.1080/02681102.2010.511702

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Quarterly*, *35*(1), 37–56. JSTOR. https://doi.org/10.2307/23043488

Sia, S. K., & Soh, C. (2007). An assessment of package–organisation misalignment: Institutional and ontological structures. *European Journal of Information Systems*, *16*(5), 568–583. https://doi.org/10.1057/palgrave.ejis.3000700

Simonsen, J., & Robertson, T. (2012). *Routledge International Handbook of Participatory Design*. Routledge.

Soh, C., Kien, S. S., & Tay-Yap, J. (2000). Enterprise resource planning: Cultural fits and misfits: is ERP a universal solution? *Communications of the ACM*, *43*(4), 47–51. https://doi.org/10.1145/332051.332070

Sommerville, I. (2008). Construction by Configuration: Challenges for Software Engineering Research and Practice. *19th Australian Conference on Software Engineering (Aswec 2008)*, 3–12. https://doi.org/10.1109/ASWEC.2008.4483184

Stake, R. E. (2005). Qualitative Case Studies. In *The Sage handbook of qualitative research, 3rd ed* (pp. 443–466). Sage Publications Ltd.

Street, C. T., & Ward, K. W. (2012). Improving validity and reliability in longitudinal case study timelines. *European Journal of Information Systems*, *21*(2), 160–175. https://doi.org/10.1057/ejis.2011.53

Strong, D. M., & Volkoff, O. (2010). Understanding Organization—Enterprise System Fit: A Path to Theorizing the Information Technology Artifact. *MIS Quarterly*, *34*(4), 731–756. JSTOR. https://doi.org/10.2307/25750703

Titlestad, O. H., Staring, K., & Jørn, Braa. (2009). *Distributed Development to Enable User Participation: Multilevel design in the HISP network*. *21*, 25.

UiO. (n.d.-a). *DHIS2 Design Lab*. Retrieved 23 April 2021, from https://www.mn.uio.no/ifi/english/research/networks/hisp/dhis2-design-lab/index.html

UiO. (n.d.-b). *DHIS2 design lab*. Retrieved 5 June 2020, from https://www.mn.uio.no/ifi/english/research/networks/hisp/dhis2-design-lab/index.html

UiO. (n.d.-c). *Health Information Systems Programme (HISP)*. Retrieved 4 March 2021, from https://www.mn.uio.no/ifi/english/research/networks/hisp/index.html

Van De Ven, A. (2007). *Engaged Scholarship: A Guide For Organizational and Social Research*.

Walsham, G. (1995). Interpretive case studies in IS research: Nature and method. *European Journal of Information Systems*, *4*(2), 74–81. https://doi.org/10.1057/ejis.1995.9

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

Xu, L., & Brinkkemper, S. (2007). Concepts of product software. *European Journal of Information Systems*, *16*(5), 531–541. https://doi.org/10.1057/palgrave.ejis.3000703

Yin, R. K. (2003). *Case study research: Design and methods*. Sage Publications.