DHIS2 Method Toolkit

Design principles for a resource supporting and promoting user centered design in a software ecosystem

Bendik Johann Kroken

Master Thesis
Informatics: Design, Use, Interaction
60 credits

Institute of Informatics
The Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

May / 2021

DHIS2 Design Toolkit

Design principles for a resource supporting and promoting user centered design in a software ecosystem

Bendik Johann Kroken

2021

© Bendik Johann Kroken

2021

DHIS2 Design Toolkit

Bendik Johann Kroken

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

Trykk: Reprosentralen, Universitetet i Oslo

ABSTRACT

The inclusion of users in the design of software is promoted by IS researchers and practitioners to help make software usable and relevant for end-users. An increasing amount of software is developed in generic software ecosystems, where one actor develops generic software and others implement this in specific organizations using the software. There has been reports of poor usability in generic software. To overcome this the organization responsible for generic design should provide resources to support and promote user centered design(UCD), as some practitioners have reported UCD methods to be costly and inaccessible. However, there is little research done specifically of how supporting and promoting UCD can be done successfully.

In this thesis, I address this gap by investigating the following research question; *What are design principles for a resource to support and promote user centered design during implementation and configuration of generic software?*

This research question is explored through a design science research project, where I have investigated the practices associated with DHIS2(generic software) implementation to identify how UCD methods can be made part of these, as well as challenges conducting these methods. Based on this insight, I have developed a UCD method toolkit that support and promote UCD in DHIS2 implementation projects.

The thesis provides the following contributions to research and practice. First, I provide five design principles for the development of a toolkit that can support and promote UCD methods in a generic software ecosystem. The principles are: 1) the toolkit should be mutable, 2) the toolkit should present a diversity of methods, processes and techniques, 3) the toolkit should present readily recognizable examples of common challenges, 4) the toolkit should present examples of solutions to challenges that are readily recognizable by designers, 5) the toolkit should have a level of abstraction that meets users' mental models. Second, the thesis practically contributes with a resource for DHIS2 practitioners to have their UCD practices supported.

Keywords: user centered design, software ecosystem, generic software, design science research, design principles

ACKNOWLEDGEMENTS

First and foremost, I want to thank my supervisor, Magnus Li, for all this help, encouragement, feedback and guidance with this thesis. Without him think thesis would not exists. Thank you for all the engaging conversations and your help to bring my project to shore in this trying year.

I would like to thank all the participants in my project, that took the time out of their busy schedules to talk to my peers and I about their work. Especially I would like to thank HISP Malawi and HISP Mozambique for welcoming us so warmly into their offices and spending so much of their time with us.

I would like to send my thanks to the DHIS2 Design Lab and all its members for all the feedback and writing sessions we had, your feedback and help has been invaluable for this thesis. I would like to thank Mats for all his help throughout this process and helping me with all the virtual interviews, design feedback and making sense of my thesis. Especially I would like to thank Anders and Hanna for being such great lab coordinators, organizing and administering all the important activates throughout this year, helping me stay motivated and focused. More importantly I would like to thank them for being such great friends, travel companions and supporters through the last year and a half, this thesis would not be possible without you and your help.

Finally, I would like to thank my friends and family for all their love and support. Especially I would like to thank Marin and Herman for being the best friends and roommates any master student could ask for. Thank you for loving me, supporting me and taking such good care of me throughout the writing of this thesis.

Thank you all,

Bendik Johann Kroken

Table of Contents

List of figures

List of tables:

List of abbreviations

DHIS2 – District Health Information System 2

HISP – Health Information Systems Programme

UiO – University of Oslo (Universitetet i Oslo)

DP – Design Principle

UCD – User Centered Design

DSR – Design Science Research

PD – Participatory Design

IS – Information System

HIS – Health Information System

UX – User Experience

ERP – Enterprise Resource Planning

EHR – Electronic Health Record

UX – User Experience

CHAPTER 1: INTRODUCTION

This thesis explores how user centered design (UCD) methods can be promoted and supported in a generic software ecosystem. The promotion of user centered design methods in this thesis is understood as how to encourage practitioners to use these methods to a greater extent. Support is understood as providing guidance in user centered methods and processes on practitioners' terms, providing them what they need to reach their goals. Specifically, the thesis reports from a yearlong Design Science Research (DSR) project, where this challenge has been addressed through design of a resource referred to as a 'design method toolkit'. The toolkit contains methods and techniques for UCD as well as commonly faced challenges to UCD and solutions to these challenges, and has been created iteratively in collaboration with practitioners in the ecosystem around a generic software called DHIS2. Based on this artefact, I derive, present and discuss five design principles for what a method toolkit to support and promote UCD methods during implementation of generic software in a software ecosystems can be. These principles are relevant for the vendors providing the generic software which has an interest in the creation of resources that support and promote UCD methods in the ecosystem their software is the basis of.

1.1 Motivation

Generic software solutions have gained traction the last decades and are implemented at a rapid rate across a diverse set of organizations (Berente et al. 2016; Grudin 2009; Pollock, Williams, and D'Adderio 2007). This type of software is not developed for a specific organization, user group or individual user but rather more generalized use cases (Bansler & Havn, 1994). The organization responsible for the maintenance and development of the generic software is referred to as "the vendor", they collaborate with third party actors (*the partners*) that make the generic software relevant for the organization the software is supposed to be used in through a process of *configuration*. This configuration is based on local requirements derived from the user organization, so that the generic software aligns with their work practices. This relationship between the aforementioned actors is conceptualized by Dittrich et al. (2014) as a *software ecosystem.*

To make software relevant and usable for end-users, the inclusion of end-users in the design process is promoted by academics and IS practitioners alike (G. Baxter & Sommerville, 2011; Grudin, 2009; Li & Nielsen, 2019; Strong & Volkoff, 2010). This is often challenging in generic software, as there are multiple user groups and organizations that use the software and the software itself not being centered on specific users (Li 2019; Strong and Volkoff 2010).

Due to a highly numerous and diverse set of users to center the generic software design on, there have been reports of usability issues in generic software (Dittrich et al., 2009; Li & Nielsen, 2019). Because of this generic approach to software design, a significant part of the design work is deferred to the partners responsible for configuration the generic software rather than the vendor providing the generic software (Dittrich, 2014). However, the process of centering the configuration of the generic software on the needs and requirements of end-users need to be supported by the vendor with resources to overcome usability issues. An example of this is the ecosystem surrounding the ERP software SAP, where the vendor (SAP), provides resources directly tied to UCD processes through their Experience (SAP, n.d.) and AppHaus (SAP, n.d.) programs, directly supporting partners to work with usability when configuring and implementing their software.

The software ecosystem that is the focus of my thesis is centered around the generic software solution DHIS2 (District Health Information System 2). The generic core of the software is developed and maintained by HISP (Health Information Systems Programme) from the University of Oslo (UiO) while local implementation teams throughout countries in Africa, Asia and Europe constitute the partners HISP collaborates with. These partners are working with the direct implementation and configuration of DHIS2 into user organizations, and are referred to in this thesis as DHIS2 practitioners. Together with the HISP vendor at UiO, they constitute the HISP ecosystem.

Within the HISP ecosystem there is an aim from HISP UiO to promote more user centered-methods in the process of configuring DHIS2 to increase usability of the implemented software. Specifically, there is an aim from the vendors side to provide resources to support the utilization of UCD methods, however there is no clear guide as to what such a resource should be. In my research project, I have collaborated with DHIS2 practitioners to understand what is required to

address this aim. Concretely, I have investigated the practices associated with DHIS2 configuration to identify how user centered methods can be made part of these practices, as well as challenges concerning conducting these methods and the current processes among the practitioners. Thus, I have collaborated with practitioners to explore resources that may support and promote the use of these types of methods when innovating in DHIS2.

1.2 Research Question

There are few guidelines on how to develop a resource promoting and supporting user centered design methods and what such a resource can be, especially in a software ecosystem. The literature discussing usability issues concerning generic software argues for the benefit of conducing user centered design methods to meet these issues, there is also some literature concerning the challenges of employing UCD methods is practice. However, this literature does not have many explicit guidelines as to how support and promotion of UCD methods concretely can be done. Providing the vendors of generic software guidelines as to how promotion and support of user centered method can be achieved when "outsourcing" usability work to partners is therefore a relevant, yet unexplored phenomenon. Further, there is a lack of prescriptive knowledge that specifically can guide both research and practice concerning what such a resource should have the appearance of.

I propose the following research question to guide my thesis:

What are design principles for a resource to support and promote user centered design during implementation and configuration of generic software?

To address this question, I have applied design science research (DSR) to construct a "design method toolkit" based on DHIS2 practitioners' challenges and practices. The resource aims to promote and support the use of user-centered methods when configuring generic software. The toolkit contains methods, techniques, and examples relevant to typical DHIS2 projects, and concrete guidelines on how to address commonly encountered challenges. The artefact developed and design principles derived from it attempts to meet the both the vendors aim to promote and

support more UCD methods being conducted during implementation of generic software, and providing DHIS2 practitioners wanting to conduct more UCD a resource to do so.

Finally, this thesis provides five design principles concerning what a resource that supports and promote user centered design in a generic software ecosystem could be. Contributing to the literature concerning promotion and support of user centered methods in software ecosystems with concrete prescriptive knowledge to meet this aim. Allowing vendors of other software ecosystems than the one surrounding DHIS2 to apply this knowledge for resource design.

1.3 Thesis Structure

The thesis is structured accordingly:

Chapter 2 – Background

This chapter introduces background information, presenting the HISP ecosystem and DHIS2. The practitioners which are the intended users of my toolkit are presented here – the DHIS2 practitioners.

Chapter 3 – Related Research

This chapter introduces two concepts important in my thesis: generic software ecosystems and user centered design. After this a presentation of the challenges with user centered design are presented as well as what ways the presented challenges have tried to be met in the literature.

Chapter 4 – Methodology

In this chapter I present my methodology – design science research. I also present my research process, the methods used to gain insight, design the artefact and evaluate it, with strengths and weaknesses of the chosen methods.

Chapter 5 - Diagnostic Findings

This is the first of two chapters presenting the empirical findings from my work. It this first one, findings from the diagnostic data collection activities that has been ongoing throughout my whole research process is presented. I will conclude the chapter with presenting design requirements

derived from analysis of the diagnostic data to support DHIS2 practitioners in their use of the designed artefact.

Chapter 6 – Design and Evaluation Findings

This chapter is presents the design process in three wide iterations of how the artifact was designed and the influences in the design. After that I present my findings from the evaluations of the artefact.

Chapter 7 – Artifact Description

This chapter presents the final iteration of the artefact.

Chapter 8 – Discussion

This chapter presents the design principles that are the main contribution of my thesis. These are discussed with my empirical findings and challenges, solutions and concepts presented and identified in the related literature chapter.

Chapter 9 – Conclusion

This chapter summarizes the thesis, the findings and the contribution.

CHAPTER 2: BACKGROUND

In this chapter I will introduce central terms used in this thesis as well as give background information on the context this thesis "exists" in. I will present three main terms that are interlinked and referenced to for the rest of this thesis; DHIS2, HISP and the DHIS2 Design Lab.

2.1 DHIS2 and HISP

The generic software, and surrounding ecosystem that is the case of this thesis is the District Health Information System 2 (DHIS2). This is a generic open source (Health Management Information Software) HMIS implemented across 73 low and middle-income countries across the globe, as seen in Figure 1 (DHIS2, n.d.; Li & Nielsen, 2019) . The generic software core is developed by a developer team being a part of HISP (Health Information Systems Programme) situated at the University of Oslo. This core team of developers are responsible for the development and maintenance of the generic DHIS2 software and is referred to in this thesis as the *vendor*.

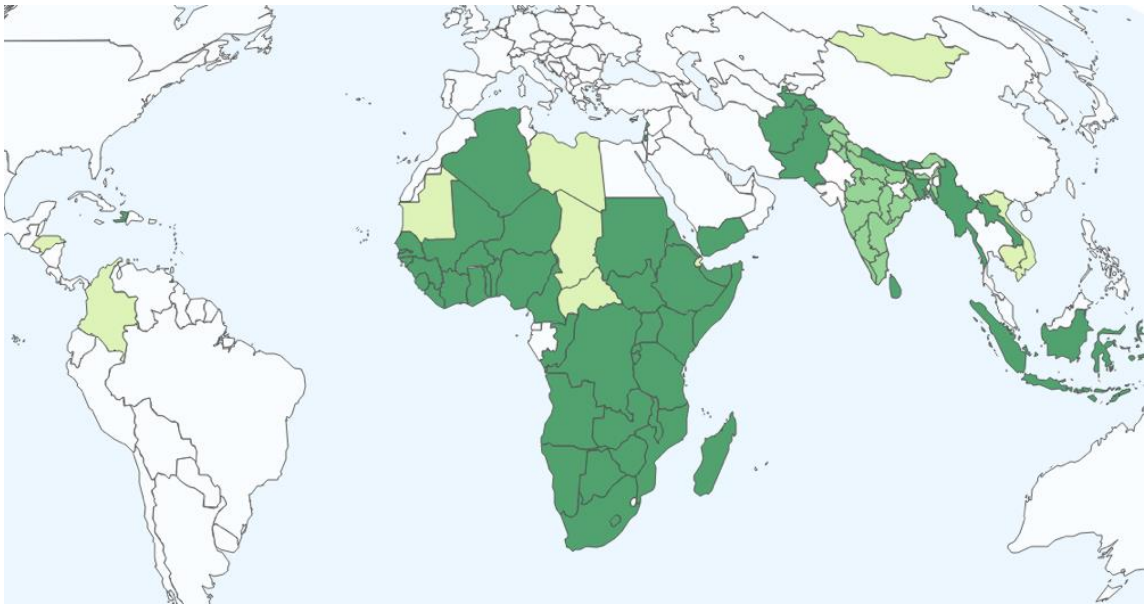


Figure 1: DHIS2 implementation across the globe (DHIS2, n.d.)

DHIS2 is used for tasks like reporting health data, analyzing and visualization of health data and logistical tasks like ordering medicine and vaccines (DHIS2, n.d.). It is used by organizations to support collection and analysis of data across a wide array of health facilities from large hospitals

to small local health clinics. Figure 2 is an example of a dashboard where different modules have been used to present health information to better make decisions.

Figure 2: Typical DHIS2 dashboard for health management (DHIS2, n.d.)

Some examples of projects are presented below. From measles vaccination tracking in Bangladesh (Figure 4) and Child and maternal health tracking in Palestine (Figure 3) to nature conservation in Eastern Africa (Figure 5)

Figure 4: Measles vaccination tracking, Bangladesh (DHIS2,2021)

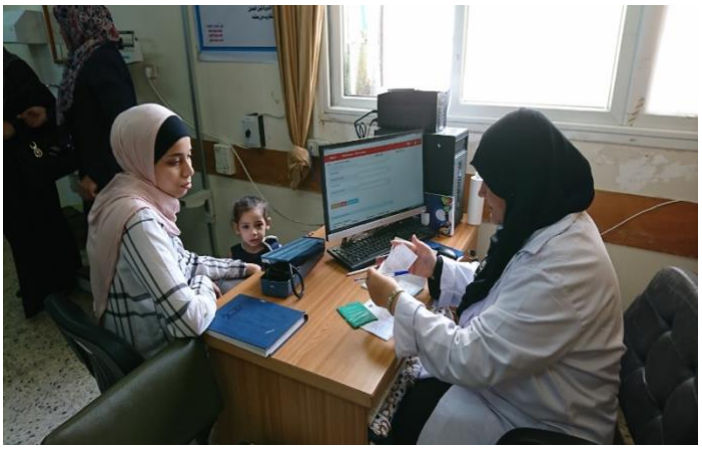


Figure 3: Maternal and child health tracking, Palestine (DHIS2, 2020)

Figure 5: Nature conservation in East Africa (DHIS2, 2020)

HISP, the organization that develops DHIS2 is made up of researchers, developers, implementers, coordinators, PhD student and master students. HISP follows a participatory approach where the aim of the organization is *"to support local management of health care delivery and information flows in selected health facilities, districts, and provinces, and its further spread within and across developing countries"* (HISP, n.d.).

HISP was established in 1994/1995 in South Africa, to support the fragmented health sector post-apartheid (Braa & Sahay, 2012). To do this, DHIS2 was developed through a participatory design project, a legacy the organization carries with it today, although its immense spread across countries and contexts.

As mentioned, HISP has a core team of developers, referred to in this thesis as *the vendor*. Which develops the generic apps and functionalities that is contained in DHIS2. The localization of the software happens across multiple local branches of HISP in different countries across the global South, these are referred to in this thesis as *HISP groups*. HISP groups are not the only ones working with DHIS2 however, and other actors not part of the organization also implement and

innovate on DHIS2, like UNICEF or USAID. In this thesis, the collective term for both HISP groups and other actors working with DHIS2 will be *DHIS2 practitioners*.

A central aspect of the DHIS2 ecosystem, is the network of different organizations and actors collaborating to provide support and capacity building for each other, and the software ecosystem as a whole. There are many different types of resources present in the ecosystem, such as documentation, implementation guides and DHIS2 Academies, capacity building events covering many different aspects of DHIS2 development and implementation (DHIS2, n.d.). The HISP groups both collaborate with each other across groups and regions as well with the development core at HISP UiO with providing requirements gathering to the generic software, feedback from implementation projects in "the field" and new innovations from the local groups (DHIS2, n.d.).

2.2 DHIS2 Design Lab

This thesis and myself are part of the DHIS2 Design Lab, a generic software design lab that is aimed at "addressing the usability and local relevance of generic enterprise software" (Li, 2019b, p. 11). The Design Lab consists of researchers and multiple master students from the University of Oslo participating in different research project related to DHIS2 and the enhancement of usability during the implementation of DHIS2 across multiple contexts (UiO, n.d.).

The DHIS2 Design Lab, is established in HISP UiO, and aims to work across both the core team at UiO as well as the implementing HISP groups. This is exemplified in Figure 6 where the core group participates in *generic-level design*, the design of the generic software and the HISP groups and other partners (DHIS2 practitioners) participate in *implementation-level design*, the configuration of generic software to fit local needs. The DHIS2 Design Lab aims to both contribute and learn from these processes, adding to a design infrastructure consisting of people, resources and practices surrounding DHIS2 (Li, 2019; UiO, n.d.).

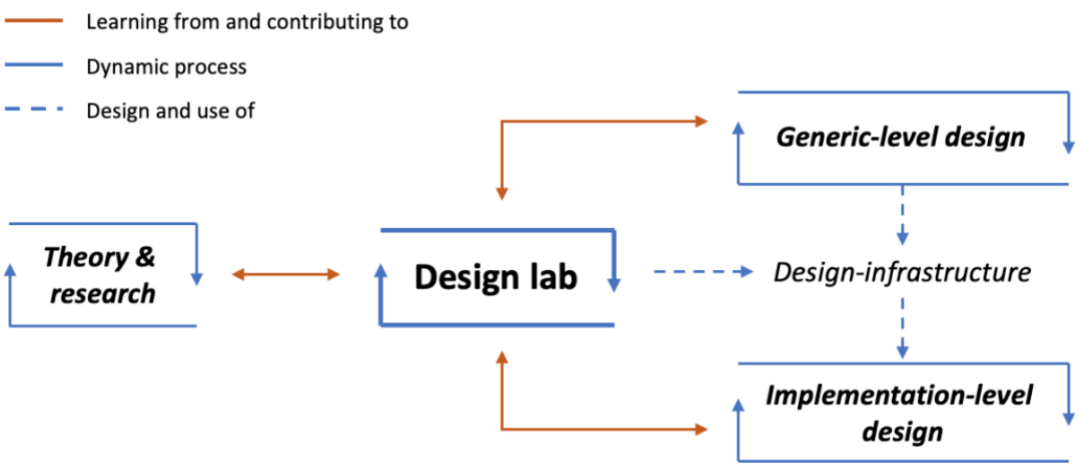


Figure 6: DHIS2 Design Lab and its role and contributions(Li, 2019b, p. 12)

The Design Lab participates in several academic activities like activities field studies in contexts where DHIS2 is implemented, like Mozambique, India or Uganda, or due to the ongoing pandemic, virtual empirical work. The lab also has activities contained to the lab itself such as workshops concerning academic writing, experience sharing and co-analysis. The participants of the lab work on different projects connected to DHIS2 like designing artefacts (like this thesis), understanding problems and practices in the DHIS2 ecosystem and planning interventions, all with the goals of addressing usability within the DHIS2 ecosystem. Within the lab there is an aim for continuity of projects, so that they do not end upon the end of master theses.

2.3 Summary

This chapter presented important terms that will be used for the rest of this thesis. Specifically, it presented the software in question, DHIS2 and surrounding ecosystem. The actors important to this thesis have been presented too, the HISP groups and other partners, as well as the DHIS2 Design Lab which my project is a part of.

Chapter 3: Related Research

The research objective of my thesis is to explore how to support and promote user centered methods when implementing generic software in a generic software ecosystem. In this chapter I will examine the existing literature concerning this topic. The chapter is structured in the following way; first I will explain my understanding of generic software ecosystems and the design processes that is involved with it, following this I will explain what I mean with user centered design. These are two major concepts I will apply as a lens in the empirical analysis. Second, the synthesis of existing literature will be concerned with challenges with supporting user centered design methods in software development practices in general, and challenges with user centered design in generic software more specifically. Third, I look at what the literature reports as ways to meet these challenges. The chapter ends with a summary and an identification of a gap in the literature.

3.1 Generic software ecosystems

This section is concerned with what a generic software ecosystem is, and how it is designed and what user centered methods are, to set the stage for the rest of the chapter.

Generic Software

Generic software is software that developed for organizations rather than specific users, for example ERP (Enterprise Resource Planning) and EHR(Electronic Health Record) software solutions (Pollock et al., 2007). These are often designed for a domain, and often multiple rather than specific work tasks. Design of generic software is different from more traditional, bespoke software projects due to it happening on two "levels" – the generic and the local (referred to as *configuration* in this thesis) (Li & Nielsen, 2019). The configuration of generic software is the process of designing and shaping it so that it becomes locally relevant for the organizations and end-users of the software. For then to be implemented into a user organization like a clinic or a hospital.

Generic software systems are designed not to center on a specific user, user organization or group, but keeping these at a distance in the design process(Sommerville, 2008). This allows the software to be used in a diverse set of contexts, after configuration for said context has taken place. The accessibility of generic software, and the reduced cost in time and development resources have made generic software highly popular in a diverse set of organizations, for example health, logistics and education (Li, 2019a; Pollock et al., 2007).

Software Ecosystem:

This thesis in concerned with generic *software ecosystems*, based on Dittrich et als (2014) understanding of a software ecosystem, simplified in Figure 6. The software ecosystem has some important aspects, that allows for collaboration and innovation. At the center, there is a software provided by a "vendor", this software is provided to third party actors to configure the software and add functionality and apps needed for the organization it is implemented into, the "partner". Due to the generic nature of software at the heart of software ecosystems, some of the design is moved from the vendor to third party actors as they are *"closer to the concrete use context*" (Dittrich, 2014, 1454), this is seen in Figure 7. This access to the core is often done through APIs and other boundary resources (Ghazawneh & Henfridsson, 2013).

A software ecosystem consists of a diverse set of actors, for the purpose of this text I am concerned specifically with the vendor, partners and user organizations (Wareham et al., 2014). As presented earlier, the vendor is the developer of the generic software itself, maintaining the functionalities and given by the software itself. The partners are organizations that are not a part of the design and development of generic software but focuses on the configuration and implementation of said software, ideally making it relevant and useful for the unique context of the user organization (Dittrich, 2014; Wareham et al., 2014). The user organization is the organization concerned with the end-use of the generic software to have their tasked supported, this is the organization in which the partner implements to software. These are the actors most relevant to my thesis and are summarized in (Table 1), but it is important to note that more actors are part of a software ecosystem, like governments (rules and regulations) which are not the scope of this thesis. Resources can also be considered a part of the software ecosystem, they can be component libraries, academy trainings for partners or documentation, all with the purpose of supporting the partners configure and implement the generic software (Dittrich et al., 2009).

Actor	Role
Vendor	Develop and maintain the generic solution. Responsible to provide resources for configuration of generic solution.
Partner (s)	Configures the generic solution to make it meet the local requirements given by the user organization.
User organization	The organization which actually uses the software for their tasks. They provide the requirements and projects the partner are a part of. These are often the end-users of the software (e.g. administrative workers at health clinics)

Table 1: Actors in a software ecosystem (Dittrich et al., 2014)

3.1.1 Design of generic software in ecosystems

Although this thesis is more concerned with the implementation and configuration of generic software, it is important to have an understanding of the design process of the vendors generic design process, dubbed *generic-level design* by Li and Nielsen (2019). This is important because it helps us understand what processes allows for local configuration and implementation to take place. As mentioned above, the design of generic software solutions is highly unspecified for user organizations. The vendor of a generic software solution gathers insight through requirements from the partners of the generic software for the development of new functionalities and maintains the core functionalities of the software(Pollock et al., 2007). Through generic design of software, flexibility is built in so that there is room at the implementation level to configure the generic software to fit the needs of the user organization(Dittrich et al., 2009), what Li and Nielsen (2019) calls “implementation-level design”. This is provided by configurability and resources to support the development of custom apps.

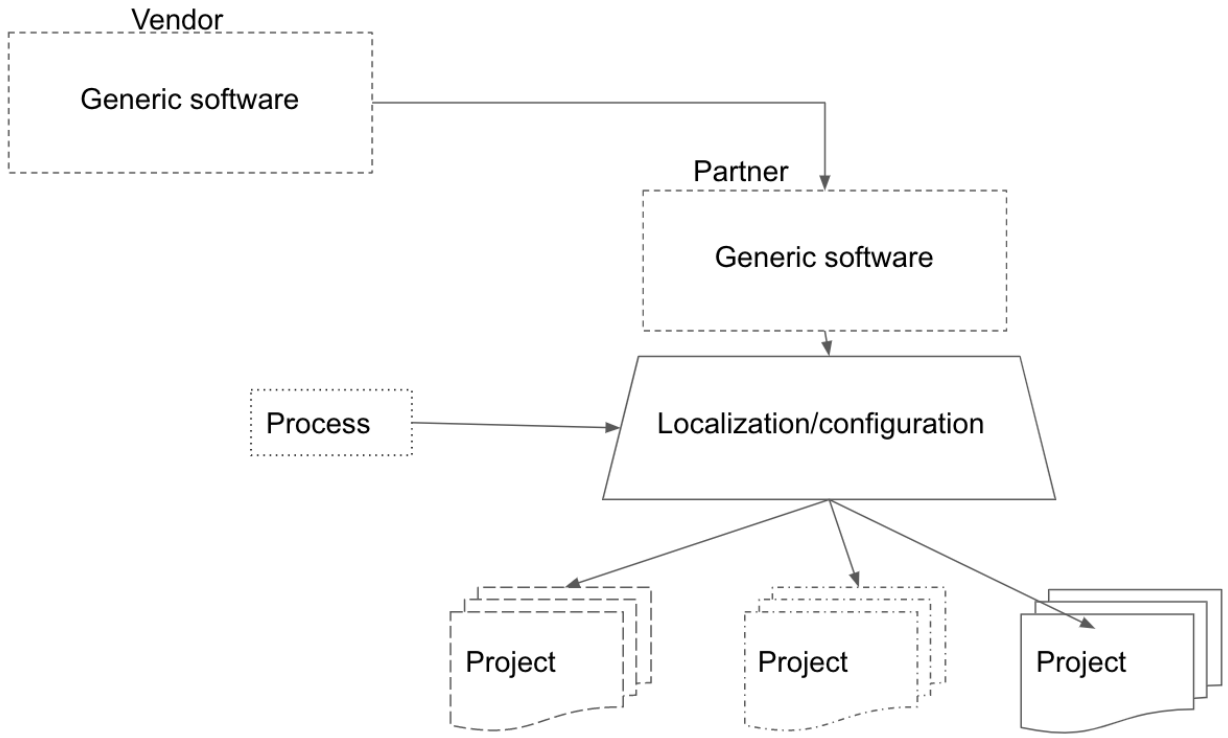


Figure 7: Localization of generic software in a generic software ecosystem

3.1.2 User Centered Design

User Centered Design (UCD) is a design approach, initially introduced by Norman and Draper (1986). The heart, namely the center, of the approach is the user of software and the needs of the user must be taken into account when developing new technology (Norman & Draper, 1986). This underlines the notion that the *"purpose of the system is to serve the user*" (Norman & Draper, 1986), which is central to UCD. Gulliksen et al. (2003) has an interesting observation concerning UCD, which unlike approaches like Participatory Design (PD), does not need to involve the user at *every* step, but rather emphasizes the importance for a deep understanding of the user and their context. Karat adds to the general understanding of UCD, *"For me, UCD defines an iterative process whose goal is the development of usable systems*" (Karat, 1997, 38).

It is important to note when discussing UCD that there is no agreed upon definition of what it is , which Gulliksen et al. (2003) argues leads to UCD being a concept with no real meaning. This

underlines the importance of understanding the main aspects of the approach rather than the definition. Thus, we can begin to understand UCD as an iterative design process with the aim of making systems to serve the users. In line with Gulliksen et al. (2003), one approach to understanding UCD is to focus on the process of conducing user centered design rather than the definition of the approach itself. This can be seen in how most UCD definitions have some returning activities which iteratively interacts with each other (Gulliksen et al., 2003; Karat, 1997; Norman & Draper, 1986; Vilpola, 2008).

3.1.2 The UCD Process

The four main activities presented by the International Standards Organization (ISO), and have been used as examples of a standardized UCD model (ISO 9241-210:2019). These are namely; *understand and specify the context of use, specify the user and organizational requirements, produce design solutions, evaluate designs against requirements* (ISO, 2018). As presented in the figure under (Figure 8), these design activities all happen iteratively and inform each other.

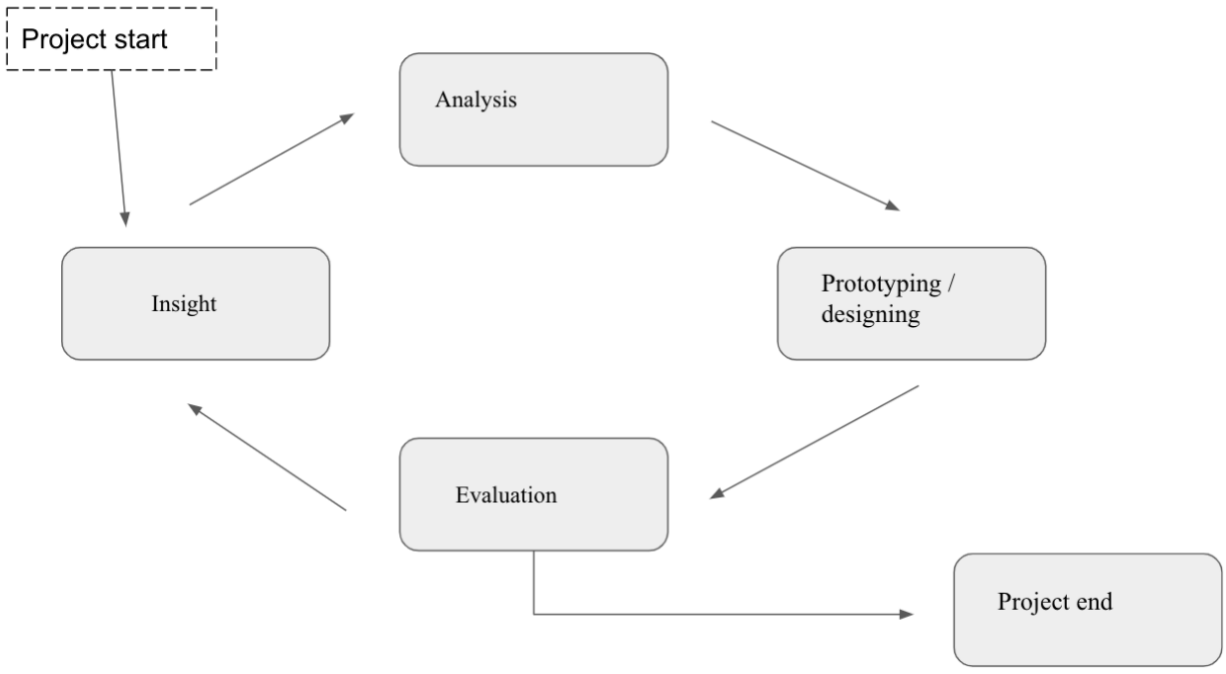


Figure 8: Typical UCD process adapted from ISO 9241-210:2019

Most typical UCD projects start with activities to develop an understanding of users and the context they work in. This phase is crucial to making sure the user is at the center of the design process and this is where the requirements from the users are gathered (ISO, 2018). During this phase, techniques frequently applied to understand user and their contexts are interviews, observations, documents analysis and workshops (Preece et al., 2015). For the rest of this thesis I will refer to this phase as *insight* or insight gathering.

Following the insight into users and their context, the designers need to analyze their findings and through this new insight specify user requirements. This is often done removed from the users either alone or in collaboration with the development team (Gulliksen et al., 2003). Several techniques are used in this phase, but affinity diagrams and brainstorming sessions are often used to make sense of the data collected (Preece et al., 2015). I will refer to this as *analysis* for the rest of the thesis.

Once user and organizational requirements are specified and derived, the development of prototypes based on said requirements follows. By designing based on user requirements, the projects become more situated in the user's context, ideally heightening the usability of the solution (Göransson et al., 2003; Gulliksen et al., 2003). Prototyping often happens on a scale from low to high fidelity, from sketches to fully realizes solutions. This is referred to as *prototyping* or *designing* in this thesis.

Once the development / design team has a prototype they want to evaluate, they usually enlist users for usability tests or take the prototypes into "the field" to gain feedback on the prototype (Preece et al., 2015). Evaluations of prototypes can go from high to low fidelity, from simple sketches to fully operational software. This allows the team to gain valuable feedback on the use of a solution, its relevance and usability. Further giving the team the possibility to make decisions on further iterations or developments of the solution based on the users themselves(Otkjær et al., 2008; Preece et al., 2015). Referred to as *evaluation* in the rest of this thesis.

The process presented in figure 7 is an abstract and idealized UCD process, and most project will deviate from how it is presented. The key aspect is that each phase informs the others, giving the design team the possibility to make the best decision for the user based on the feedback the gain

(Preece et al., 2015). This is ideally an iterative process as well, where the iterations will "go" until the requirements are fulfilled leading to the projects end. It is important to note that most requirements are not static and will change across space and time, adding to further complexity throughout this process (Mithun et al., 2018). The strength of an iterative process like the one described above is that it allows for testing multiple prototypes, tailoring it for the context of use. This allows for the software solution to be based on the needs of the users and the organization the solution is designed for (Gulliksen et al., 2003)

As presented in the previous sections, user centered design is an iteratively design approach which focuses on the users, their needs and their context. The approach has proved to be successful in increasing usability of software solutions and is starting to be popular in the IT industry, but still not applied to any large extent (Ardito et al., 2014; Bang et al., 2017; Choma et al., 2015; Mithun et al., 2018; Vilpola, 2008). Presented earlier, Gulliksen et al. (2003) argued for the lack of a definition of UCD due to there being to many rending the term meaningless, and rather focus on principles and process, two interlinked aspects in UCD.

3.2 Challenges with conducting user centered design in software projects

The following section discusses and reviews the challenges presented with conducting user centered design in software projects. Although usability enhancing, these types of approaches do have challenges and difficulties associated with them. Specifically, findings from research concerning promotion and support of these types of methods are presented. This research is further leveraged upon during the formulation of design principles later in this thesis. Through the focus on challenges with conducing user centered design I aim to be able to develop more specific solutions to these challenges, and be able to discuss these with my own empirical findings. The presented design principles will be an effort to try to meet some of these challenges.

Research centered around the question of why large, generic systems fail, have presented methods for inclusion of users and lack of systemic use of these as a major issue , especially focused on the complex socio-technical setting (Baxter & Sommerville, 2011;Dittrich et al., 2008b; Dittrich, 2014; Mumford, 2006). Scientific studies have further revealed that these types of methods are not

being applied in software development practices to any large extent in software companies, although the proven utility for dealing with usability issues (Bang et al., 2017; Brhel, 2015).

Promotion and support of user centered methods have been explored through a handful of studies conducted within software companies (Ardito et al., 2014; Bang et al., 2017; Wong et al., 2016). From existing literature, I have identified three main challenges with concerned with conducing user centered design in software projects, namely a lack or willingness to spend time and resources on these methods (Ardito et al., 2014), "developer mindset" (Otkjær et al., 2008) and their lack of inclusion in projects calls for tenders(Ardito et al., 2014; Wong et al., 2016).

Time and effort (resources)

Time and resources are often cited as the major hurdles for the use of user centered methods (Bang et al., 2017; Cornet et al., 2020; Hartswood et al., 2002; Otkjær et al., 2008). User centered methods are perceived as "heavyweight" (Teka et al., 2018) by being too time and resource consuming, which Baxter and Summerville (2011) argues are demotivating companies to apply user centered methods to any large extent. This is echoed by both Ardito et al. (2014) and Otkjær et al. (2008) which have done extensive research into the use of UCD in IT organizations. Otkjær et al. (2008), investigated the origin of the perception of the costliness of these types of methods, and none of the participants had any clear explanation to why this is. This could be stemming from a lack of familiarity with methods, and as the authors report , there is widespread lack of understanding of usability and the process related to usability work across organizations(Otkjær et al., 2008). This finding is mirrored by Ardito et al (2014) as they argue the lack of UCD professionals or general UCD expertise makes finding methods that are relevant and with fitting use of resources challenging or an additional expense needing to hire external support to conduct or lead design processes.

Otkjær et al. (2008) reports a challenge is to have people participate, as it hard enough for firms to motivate clients and their users to evaluate the whole system. This process takes a long time and they use a significant amount of resources on it, underling the need to find methods that work in specific contexts.

"Developer mindset"

Otkjær et al reports this is a significant hurdle of user centered methods, as some developers argue that they are unable to "think like users" (Otkjær et al., 2008, p. 27). This finding is further elaborated and echoed by Ardito et al (2014) as they report that many developers are fully focused on programming and other technical aspects, lacking knowledge about usability, UX and methods for a user centered design process. Further, the authors report that many developers do not view the methods promoted relevant to their processes or domains they work in (Ardito et al., 2014). The value of user centered design methods is closely related to this aspect, as the "developer mindset" encompasses a lack of value afforded to the continuous and active involvement of users throughout the process (Ardito et al., 2014; Otkjær et al., 2008; Teka et al., 2018).

Developer mindset was further explored by Teka et al. (2018), which explored UCD methods and promotion in the Global South. They argue that the lack of professionals trained in UCD methods is another reason for the lack of importance that is given to these types of methods. One argument was that traditional UCD methods lack consideration of *"complex socio-economic conditions, lack of infrastructure and cultural heterogeneity of developing countries"* (Teka et al., 2018, p. 36).

Not included in calls for tenders

Ardito et al. (2014) has an interesting finding that is not explicitly mentioned in much of the literature around the promotion of UCD methods in that these types of methods and methodologies often are not included in calls for tenders. The calls for tender is important in any software project, and it is this document that "sets the stage" for the project to follow, thus if user centered methods are not included, they have less importance for the client leading to a lack of priority for the software company. This phenomenon was demonstrated by Martin et al. in their ethnographic investigation of designers on an implementation team of an ERP projects, where usability activates were constantly being removed and deprioritized for *"more important priorities than usability"*(Martin et al., 2007, 254).

To meet these challenges, many of the authors presented argue for the need to adapt methods for the unique projects and practices across organizations. This is even more highlighted by Teka et al. (2018) discussing cultural sensitivities and digital divides between rural and urban areas when working with UCD processes. It should be acknowledged as well that UCD methods are deeply rooted in a Scandinavian tradition (Mumford, 2006) where community structures often are quite different from many of those in the global South (Teka et al., 2018), further underlining the need

for methods to be contextualized to be aware of local conditions. Therefore, by promoting and supporting local practitioners and giving them the tools to work with methods suitable for their context becomes crucial (Teka et al., 2018).

The presented challenges are summarized in Table 2.

Challenge	Description	Example reference
Time and resources	UCD methods are perceived as heavyweight and resource draining.	(Bang et al., 2017; Teka et al., 2018)
"Developer mindset"	UCD methods are not seen as important. Users are seen as challenging to work with.	(Otkjær et al., 2008)
Not included in calls for tender	UCD methods are not included in calls for tenders, signaling that they are not important or prioritized.	(Ardito et al., 2014)

Table 2: Commonly reported challenges with conducting UCD in software projects

3.2.1 Challenges with conducting UCD during generic software implementation

Not designed for a specific user/user group

Developing generic software by nature makes it very hard to involve end-users to any great extent, as the generic solution developed by the vendor is not supposed to be for one homogenous user (or user group) (Li & Nielsen, 2019). The software itself needs to serve a diverse set of users and uses, a process that makes it challenge to tailor software to users and their local needs (Li 2019). Due to this development practice, most generic systems needs some type of customization or localization when it is to be implemented in an organization to be usable (Dittrich et al., 2009b).

As Teka et al (2018) notes, a lack of professionals that can perform usability tasks such as evaluations is a significant challenge in the application of user centered methods. Thus, the

usability work falls to a group of professionals not trained in these types of methods, nor motivated in doing them.

Generic systems, are by nature different form that of bespoke projects, underlining the need for "other" methods. Approaches like participatory design does not "fit" successfully in development of generic systems due to the nature of the system, due to *"design in these areas takes on a very different meaning than in the more traditional design paradigm[..]"* (Simonsen & Robertson, 2012, p. 50).

The software itself and the degree of flexibility provided from the vendor becomes highly important too in a user centered approach as this quickly becomes one of the limiting factors for user centered design. The more flexibility the vendor provides, the more space there is for the involvement of users and flexibility of methods. As Li (2019) notes, there are specifically two main challenges to conducting user centered design during the implementation of generic software; the software needs to fit many contexts and purposed and the design of the generic software itself presumes a top-down approach with pre-defined UIs and functionalities.

3.3 Efforts to meet UCD challenges when conducting software development

There have been several efforts to meet the challenges of conducing UCD methods presented over by an array of authors. Discussing approaches to the support and promotion of more user centered methods, Baxter and Summerville (2011) introduce the term *method usability* when discussion why some methods not always work. This concept is concerned with how relevant the method itself is to the context it is to be used in and the ends meant to be achieved by the method (how relevant it is to practitioners). However, as becomes apparent in much of the literature concerning user centered methods and their utilization, *"[…]methods mostly provide advice for sympathetic systems designers rather than detailed notations and a process that should be followed"* (Baxter & Sommerville, 2011, 7). Acknowledging the notion of increasing method-usability, Li and Nielsen (2019) calls for an increased focus on how to increase usability not only by providing technical flexibility and support of the generic software itself, but also support for methods achieving usability underlining a holistic approach to usability. Working in a complex socio-

technical context, as many of the studies under do, ethnographically inspired research methods have been proven useful as seen in Ardito et al., (2014), Dittrich et al., (2009) and Holtzblatt et al., (2005)

Collaboration and communication

Collaboration and communication to develop unique methods for specific contexts and projects have been explored through cooperative method development (Dittrich et al., 2008) and co-realization (Hartswood et al., 2002). Both these approaches have as a focus to collaborate with practitioners and asks for a long engagement in the development context. Central to Cooperative method development is the creation of methods specifically for a domain to meet local challenges, and these methods are developed in collaboration with practitioners in said domain (Dittrich et al., 2008).

Overcoming the hurdle of poor communication between development teams and users in the configuration of "off-the-shelf" software, Hartswood et al (2002) proposes c*o-realization* as a potential practice to do this. Central to Hartswood et als (2002) notion of co-realization is long engagement, allowing the developing team to both fully understand the practices of users as well as allowing users to grow into a system making it minimally invasive to the work practice (Hartswood et al., 2002). Co-realization and cooperative method development both are heavily depended on ethnographically centered methods in that is a time-consuming endeavor as well as needing a high level of skills from practitioners for it to be successfully executed. Both these being challenges previously mentioned by authors like Bang et al. (2017) and Ardito et al. (2014) when trying to apply these methods in software projects in the IT industry.

Time and resources

Time and resource costs are frequently the most noted challenge of applying user centered methods in software projects. A solution to meet the need for time and resources to conduct user centered methods, especially in more corporate (non-academic) settings, have tried to be found by an array of authors.

Holtzblatt et al (2005), argues for the need for rapid user centered methods, as this is highly relevant not only due to time and resources used, but also due to the fact that software technology moves at a very fast pace (Holtzblatt et al., 2005). Issacs (2012) demonstrated the applicability of

method this in her study of parking systems, where she applied a "rapid ethnography", that was reported as "satisfactory for clients". A rapid ethnography is an ethnography that includes all of the typical activities of an ethnography in the fraction of the time to gain insight into users work practices (Isaacs, 2012).

"Discount usability methods" have been promoted in literature for a possible solution for the lack of time and resources reported by the IT industry about the lack of applying these methods (Göransson et al., 2003; J. Nielsen, 1995). At the heart of these methods is the premise of *"some usability is better than no usability"*(Cockton & Woolrych, 2002, p. 14).

By applying low-cost techniques for design and testing, this approach aims to not find the very best system but to design a system that is productive and satisfactory enough for the user (Kane, 2003). Examples of these methods are scenarios, thinking out loud, heuristic evaluation and general common design principles, meant to be accessible, have a "low bar" and are described as *"quick and dirty"*(Kane, 2003; J. Nielsen, 1995; Jakob Nielsen, n.d.).

Coined by Jakob Nielsen (1995), he wanted this approach to user centered design to be done by everyone in a development team, thus presenting these methods that takes little time and effort, has tangible results and seem accessible and relevant to practitioners. This approach does not come without its critique, as Cockton and Woolrych (2004) notes, the discount usability methods can only consider a few factors at a time (since that is all you have time for!). You use "smaller" methods for larger, often abstract problems, reducing the design space – not allowing for the solutions the users really need to be instantiated. The authors argue instead that instead of "discount methods", all user centered methods should be strengthened *"so that discount methods are less discounted and "full strength" methods can be applied in more contexts"* (Cockton & Woolrych, 2002, p. 17).

Otkjær et al. (2008) investigated specifically the argument that time and resources were too draining, as they note that frequent interactions with user have showed that overall costs for projects go down. In their study they prove that software companies do not have any clear cut reasons how exactly including users increase the time and resource cost(Otkjær et al., 2008). Which in turn might mean that a lack of understanding of what user centered methods are, and a continuous reproduction of the myth that it is resource draining and costly.

Resources

The literature around the support and promotion of user centered methods and approaches centered on specific resources in IS is scarce, underlining the need for more tangible and accessible resources. Some authors have tried to promote and support the use of user centered methods through resources, as they allow the user to be supported in their processes while maintaining autonomy, selecting the methods and approaches deemed relevant to their context and projects (Cvijikj & Michahelles, 2011; Heer et al., 2005; Piller & Walcher, 2006). Bang et al. (2017) notes in their article that resources for support and promotion of user centered methods is especially important in organizations where the practitioners have limited experience with these types of methods.

Howarth et al. (2009) developed in their study a tool for usability engineering practitioners to support them during their processes when encountering usability problems. This tool was centered on problem instances and solutions to these, and was proven useful for both novice and senior usability engineers (Howarth et al., 2009). This approach to tool design helped as a form of scaffolding, as the users of the tool experienced familiarity with situations and problems presented. The study was a closed-usability evaluation, thus the use of such a tool in a more complex-socio technical context is yet to be explored. A solution to the question as to how to support user centered methods, Bang et al argues for *"not just talk about the methods, show the procedures and results"*(Bang et al., 2017, p. 184). By doing this, the authors argue that practitioners have a more tangible artefact to relate to, especially in industry (as opposed to academia), where there is a more result-oriented approach to methods and different approaches. Like the study done by Howarth et al. (2009), Bang et als (2017) study was also done with smaller companies working on bespoke software projects, underlining the need to explore these ideas in generic software ecosystems.

Li and Nielsen (2019) argues for the establishment of a robust *design infrastructure*, a collection of resources like design-methods, implementation-guidelines and documentation to help implementation specialists configure and localize generic software to fit local needs. A resource made for the support of user centered methods in a generic software ecosystem does therefore need to not only fit into a local context for design, but a design infrastructure as a whole. This is proposed by Gøransson et al. (2003) in their investigation of usability tools, as they argue *"usability tools*

must be integrated in and relate to the software-development process"(Göransson et al., 2003, 115)

3.3 Literature Gap

Usability issues are not new in IS literature, whether it is concerned with generic systems or smaller, bespoke software projects. Like mentioned above, the major solution to this is argued to be the adaption and use of methods centering on users, their practices and their needs. However, in the literature there is little research in how this process can be supported by the vendor in the most efficient way, nor are there many practical examples of UCD methods implemented (Cornet et al., 2020). This especially rings true for larger, generic systems and ecosystems they are a part of. Returning to Baxter and Summerville (2011), they call for specific guidance and advice to how to select, apply and support user centered methods when developing software.

Thus, the contribution of my study will be to address this theme by filling this gap in the current literature. Specifically, in presenting design principles for developing a resource (in my case a toolkit) to support user centered methods during the implementation and development of generic software. The principles are specifically focused on the utility of the artefact and how to make user centered methods relevant for local practitioners. This artefact is developed to address the question of how, why and when practitioners should and can utilize user centered methods to try to meet usability issues.

CHAPTER 4: METHODOLOGY

In this chapter I will present the research methodology of my thesis; design science research and how I used this to develop the artefact and derived the presented design principles from this instantiation. The principles are developed and presented to support user centered methods generic software ecosystems, the principles are derived from an instantiation of the designed artefact, a design method toolkit.

The objectives of my thesis are to support and promote user centered methods, as this is an aim for the vendor of the software ecosystem I am studying – DHIS2. This is done through the development of a resource for support, and my aim is through this process to present design principles that are based in empirical evidence, design insight and theory. The empirical evidence is based on the experiences and insight of DHIS2 practitioners, which are the intended users of the DHIS2 method toolkit.

4.1 Choice of methodology

Initially, my study was to be an action design research study, where the aim was to join a development team and participate in projects with the HISP Mozambique group and in this way work locally to promote and support user centered design. My interest during the initial visit to Mozambique was to investigate current practices and projects taking place, so I could return to the group in the Fall of 2020 to build, introduce and evaluate an IT artefact this organizational setting (Sein et al., 2011). An initial aspect of an action design research study is to "diagnose" a social situation through analysis and understanding the context of practitioners, which the field study functioned as. Another important aspect of the field visit in January 2020 was also to establish relationships and get to know the local DHIS2 practitioners in Mozambique which I have maintained through informal and formal communication throughout my thesis.

Due to the ongoing covid-19 pandemic rendering travel impossible I had to approach my study in a new way. I was still interested in practices and the promotion of user centered methods for design, and the issue of how these methods could be supported was still highly relevant, I just had to approach the matter in a new way. Keeping with the original problem of how to support user

centered methods, I changed my approach to create an artefact, *the design method toolkit*, to keep exploring the research problem. Although remotely, I kept in contact with local HISP groups as well as approaching new groups to inform my study. Through the creation of the toolkit, I changed my methodology from action design research to design science research (DSR). As a methodology, it allowed me to keep working remotely with the HISP groups, though the focus has changed from designing an artefact and introducing it to an organizational setting to focusing only on the design of the artefact. This was possible to do from Oslo, where I was able to supply information through online interviews and workshops to gain further insight into practices and the challenges or new solutions DHIS2 practitioners meet.

4.1.2 Design science research

Design science research has a focus on practical relevance as well as scientific rigor (Baskerville et al., 2018) which fits well with my thesis where the objective is to explore resources to support user centered methods in DHIS2 (useful artefact) and through this I can provide design theory for the creation of such an artefact in the form of design principles (scientific rigor). Hevner and Chatterjee notes that design science research also should be motivated by *"[..] improve the environment by the introduction of new and innovative artefacts and the processes for building these artefacts"*(A. Hevner & Chatterjee, 2010, p. 17). Thus, DSR becomes an essential part of IS research since the IS field should not only *"try to understand how the world is, but also how to change it"* (Carlsson et al., 2011). This is important, as it underlines one of the most central aspects of design science research, the aim to establish utility rather than truth, in other words, it is concerned with what "works" rather than what is true.

Design Science Research is a methodology where the research is done through design of an artefact, and producing prescriptive knowledge in the shape of design principles (Gregor & Hevner, 2013). The design is often understood twofold, as a process and a product (Hevner et al., 2004). The process being the design activities and what informs these, and the product is the artefact produced through design activities. An artefact is in its broadest sense something that is manmade(Gregor & Hevner, 2013), and in DSR is takes the shape of models, methods and tangible IT systems (an instantiation) (Hevner et al., 2004). A central aspect of DSR is also the possibility of cyclical development, a crucial characteristic of design which allows for the iterative design of

the artefact, as well as connecting that design to a theoretical aspects or literature (A. Hevner & Chatterjee, 2010).

As a methodology in IS research, DSR is also concerned with the intersection of technology and organizations and as Hevner et al. notes: "*Design science, as the other side of the IS research cycle, creates and evaluates IT artifacts intended to solve identified organizational problems"*(Hevner et al., 2004, p. 77). The goal of DRS is to produce knowledge which is prescriptive and an artefact with has proven utility for the practitioners you aim to solve some organizational problem for (and with). The involvement of practitioners and "real life" problems is therefore central to the pragmatic approach of DSR. The other goal of DSR is to provide prescriptive knowledge in the form of design principles, through these making design knowledge available (Möller et al., 2020). Thus, the artefact developed is supposed to act as an instantiation of prescriptive knowledge that can guide action (Baskerville & Pries-Heje, 2010; Hevner & Chatterjee, 2010; Seidel et al., 2018).

4.2 Research Process

Several researchers have proposed process models for design science research(Baskerville et al., 2018; Gregor & Hevner, 2013; A. Hevner & Chatterjee, 2010; Peffers et al., 2007; Venable et al., 2016). However, due to the iterative and pragmatic nature of my study, it was hard to follow a set model. The importance of iterations and several cycles however was absolutely present in my study. One can identify three main, broad activities of my study;

1) Diagnosing, 2) designing 3) evaluating, as presented in Figure 9.

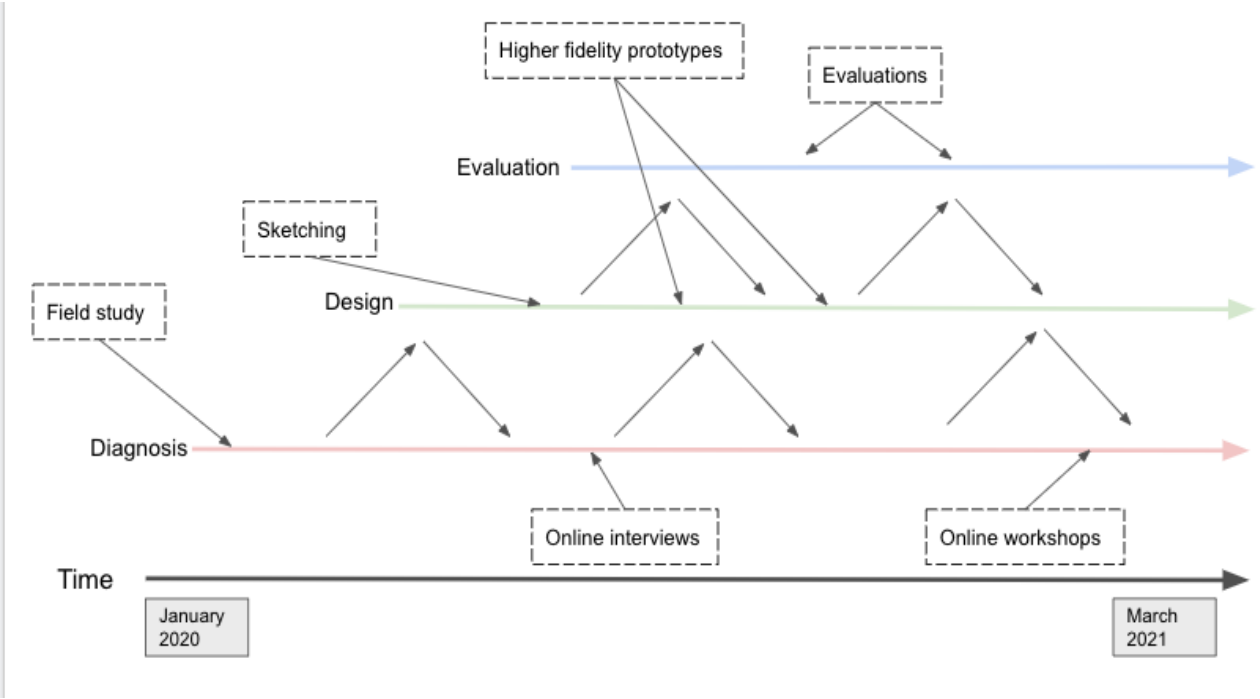


Figure 9: Research process with exemplar activities

The model above (Figure 9) illustrates my research process as work in one activity informed and prompted move in another. The organic nature of my project, and as a project part of the DHIS2 design lab the activities does not have an end or start, but rather gets "activated" by a move in another activity. As for example a design decision would lead to a need for further diagnostisation or evaluation.

Three activities guided my empirical work; diagnosis, design and evaluation (Table 3).

Activity	Learning goal	Example of method
Diagnosis	Understand practices and subsequent challenges in these.	Online interviews, contextual inquiry, observation, focus groups, workshops.
Design	Explore solutions, instantiation of design principles	Prototyping, sketching, researching toolkits

Evaluation	Evaluate artefact, explore utility, feedback from practitioners	Demonstration, walk through

Table 3: Empirical activities

Broadly speaking, as presented in the table above (Table 3), *the diagnostic activities* were concerned with understanding the practice of DHIS2 practitioners. DSR literature notes the importance of relevance for "the real world" in DSR as the aim is to solve problems through the construction of artefacts (Gregor & Hevner, 2013, 2013; Hevner & Chatterjee, 2010; Peffers et al., 2007). *Design activities* is where the development, conceptualization and instantiation of the artefact takes place, the core of any design science research study. Through *evaluation*, the artefact is demonstrated to the intended practitioners to determine the success of design principles and the utility of the artefact itself. All the knowledge and insight these activities brought has together formed the combined basis for the *justificatory knowledge* of my design principles. In this thesis, the term justificatory knowledge is in line with Gregor and Hevners understanding as *"[..] any knowledge that informs design research, including informal knowledge from the fields and the experience of practitioners"* (Gregor & Hevner, 2013, p. 340).

4.3 Methods for diagnostic work

As noted in the previous section, the first phase of my study started broad for the formulation of problems (Peffers et al., 2007). It is important to note that at this point, my aim was still to do an action design research project, where I wanted to identify points for intervention in local practice.

The general aim for the diagnostic methods were to understand the work practices, the current challenges, the solutions to these and find ways to promote and support user centered design to meet challenges and change practice. To gain this insight I needed to apply a series of qualitative methods to understand what, how and why practices are as they are. This insight was used to derive design requirements from DHIS2 practitioners to understand what content is needed and what form this should take.

4.3.1 Understanding the context

Contextual understanding was key in my study, as I aimed to work with practice and the tacit knowledge that exists in a complex socio-technical context. The term contextual understanding in this thesis is used to explain the processes the different HISP groups have for configuration and implementation as well as the projects, the people involved in these and especially where the interaction with end-users takes place.

A prime example of this was during a field visit to Malawi and Mozambique in 2020. Myself, my supervisor and my fellow master students in the design lab were able to visit the HISP teams in these countries, getting to know them and gaining insight into projects, work practices and the context these exists in. This was an incredibly valuable trip, as it helped us understand so much more about practices and who the people being a part of the HISP groups were, getting to know them, and them us.

4.3.2 Field-study: Malawi and Mozambique January 2020:

The following section explains the methods used to gain insight into practices of HISP groups in Mozambique and Malawi January 2020.

Field visit Malawi

During our field visit in Malawi, my fellow master students and I participated in meetings with the local HISP team in Zomba. Much of the data gathering was centered on gaining insight into the process of developing and implementing DHIS2. The HISP team in Malawi had a keen interest in learning more about UCD processes and methods for conducting this in their local context. It was of interest to understand the practices of the teams and therefore their general development models to potentially understand the challenges they experience and the potential solution to these. We were welcomes into the offices of HISP Malawi and participated in meetings with members of the team, introduced to the projects they participate in and what challenges that currently are present.

Figure 10: Group interview Malawi

Activities	Participants	Learning goals /key insight
Informal interviews	Members of the HISP Malawi team.	Understand process, projects and problems
Observation	HISP Mozambique office	Understand in-house practices around software development.
Group interviews	Members of the HISP team	Understand organizational map and organizational hierarchy

Table 4: Empirical activities Malawi 2020

Field visit Mozambique

Following our visit to HISP Malawi, my fellow master students and I were invited to visit HISP Mozambique. Our days during the visit was much like those in Malawi, although we were a longer time in Mozambique, having more time to get to know the practitioners, their projects and their practice. We participated in meetings, had group interviews and observed their day to day practice

working with DHIS2. I was able to join a few of the members of the DHIS2 team in a field trip to another municipality, gaining very important insight into how the team works directly with users in the field, elaborated below.

Activities	Participants	Learning goals /key insight
Informal interviews	4 members of HISP Mozambique office.	Understand process, projects and problems
Observation	HISP Mozambique office.	Understand in-house practices around software development.
Group interviews	Members of the HISP team, both senior and more junior members.	Understand organizational map and organizational hierarchy
Contextual inquiry	3 senior members of HISP Mozambique.	Understand user centered methods in action (in the field)

Table 5: Empirical activities Mozambique 2020

Methods for data gathering during field visit

Throughout our field visit in January 2020, my fellow master students and I used a diverse set of data gathering methods to gain rich insight into the practices and challenges of the HISP groups we visited. These are elaborated in the following section.

Interviews

Throughout the field visits in Mozambique and Malawi, me and my fellow master students conducted semi-structured interviews to understand practice. As a method, this is especially successful to help the practitioners reflect on their practice and why they are doing it (Edwards & Holland, 2013). For our purposes, we wanted to know more about a typical research process and the types of projects taking place in the HISP groups. The interview functioned as a good introduction to the manner and to give us a "language" to understand the findings through and discuss these with practitioners.

As seen in the tables above (table 4 & 5), I participated in two types of interviews. Group interviews and informal interviews. Much of our time visiting the HISP offices in both

Mozambique and Malawi were spent in meetings with various members of the HISP teams as well as us. These are what I refer to as *group interviews* in the tables above. The strength of these interviews was that many different people participated, providing a diverse set of views and opinions, and we could all discuss of each other's points. This helped the practitioners that participated too meet somewhat of a consensus when discussing abstract aspects like development processes and methods that took place. Usually, my fellow master students and I would come prepared to these group interviews with themes or wide questions we wanted to discuss or clear up from previous meetings and days.

Informal interviews were important as they allowed me to have spontaneous conversations and gain insight concerning phenomena that suddenly emerges. These types of interviews are referred to as ethnographical interviews by Edwards and Holland (2013), as they do not have the preparedness present as a typical semi-structured interview and relies to a large extent on the skill of the interviewer themselves. The informal interviews were more often than not causal conversation about an observation made or practitioners chatting about their day to day work, but were imperative for me to understand the logic and ideas behind the observed behaviors.

Observation

Observation was one of the most important data collection methods in the beginning of my study. This is due to the tacit knowledge concerning practice. A general challenge of investigating practice is just this, something you do which you often do not have the words to explain(Visser et al., 2005). Observation lends itself, especially in combination with interviews, to help discuss and understand practices.

Figure 11: Observation Mozambique

To gain insight concerning practice and the context of practice, myself and fellow master students observed the DHIS2 practitioners we visited working in the office and during meetings. This gave us insights into who does what work, new questions for why and how something is done and the general tools and practices to support these.

Contextual inquiry / participant observation

During the field visit in Mozambique, I was able to join some members of the HISP team during a field visit to understand how they work with users in the field. This has been especially important in my understanding where there is room for design and interventions. I was with the team from the HISP group for 3 days in the field, visiting clinics, markets, educational facilities and participated in meetings with different officials. Example projects were a market taxation project the major of the town wanted to move beyond the current pilot project at place. Another type of project was to check in on previously implemented projects, like the use of DHIS2 at health clinics.

Figure 12: Field visit

In contextual inquiry the researcher is supposed to take on a role of an "apprentice", and the participant or subject of the study the "master" (Chafi, 2020). This is an appropriate description of the field study, as the team thought me what they were doing, why and how, especially if we encountered obstacles I was told how to deal with these. The initial thought was also for me to return to Mozambique the following fall to have a more active participation in these types of activities, thus they were "training" me to be a part of the team.

This was an especially useful method for understanding practices, as it allowed me to immerse myself more in the practices and context these practices exist in. It was crucial to understanding how user centered methods were applied in the field. Like stated in the previous section, it was useful to see these methods as there for examples is very little user interaction at the HISP office. During informal conversations between stops and activities I had the opportunity to get to know the HISP team as well as asking about the logic and reasoning for different activities like photo and film documentation of the contexts we investigated

Another highly important aspect of being a part of the HISP team in the field, was to understand all the contextual and *"infrastructural issues"* (quote from HISP implementer). Coming from a completely different background and cultural context, being able to position the experiences and stories shared with myself and other master students were very useful. Being able to see the tacit knowledge in action was also highly beneficial for me, as talking abstractly and theoretically about practices and methods can only help you understand these practices so far.

Co-analysis with practitioners

An important aspect for me and my fellow master students during our field trip to both Mozambique and Malawi was to present and discuss our findings. This was especially important for us to have the validity of data "checked". Through a co-analysis we were able to clear up any misunderstandings, both semantic and practical. The co-analysis sessions were also successful in specifying my focus and what is possible with practitioners (especially with an understanding of interventions).

Much of the analysis work that has taken place during this master thesis has been co-analysis, either with participants or with fellow master students.

While conducting field work in Mozambique and Malawi, my fellow design lab members and I would present our findings, analysis and understanding of practices to the DHIS2 practitioners we would be working with. This was so that we could "test" our findings, and therefore make sure we understood the context and practices in accordance with the participants.

4.3.2 Virtual data gathering

Due to the ongoing pandemic, much of the data collection and diagnostic work had to be done virtually, as field visits were no longer possible. There were two major types of data collection taking place, online interviews and online workshops. These were done though the videotelephony software "Zoom".

Online interviews

In the fall of 2020 I conducted several online interviews with DHIS2 practices in Malawi, Mozambique, Uganda, Tanzania and Canada/US, all with experience of developing and

implementing DHIS2. These interviews were long (2+ hrs) and were focused around specific learning goals rather than set interview guided with written out questions. This allowed a fellow master student and myself to be more flexible in our interview and making sure we were learning about the themes we were interested in. Before we conducted these interviews, we would send e-mails to the participants, with some open-ended themes and questions so the participants could prepare, as well as information concerning recording and other practicalities.

The aim of these interviews was to refine what I learned during the field visit, and get in depth information about projects and projects and their challenges and possibilities. It was useful to conduct interviews as they helped me focus my thesis after the change of methodology, theme and focus due to the pandemic.

Through some contextual information like gestures and physical reactions were lost and sometimes misunderstandings and bad Internet connections made this a challenging approach to data gathering. It has some strengths I would not be able replicate otherwise. The strengths of doing online interviews were:

- Not geographically confined, I was able to include more perspectives and a more diverse set of practitioners. This was important as I did not have geographically confined the users of the resource.
- I was able to do full recordings, something I did not do of the conversations and interviews during the field work as it did not feel appropriate.

The online interviews happened very much in tandem with the design work of my artefact, one interview informing the next design decision and vice versa.

Online Workshop

In the spring of 2021, I had the opportunity to participate in workshops with other master students in the design lab focusing on challenges and possibilities for more user involvement when developing DHIS2. The participant groups were already members of HISP, and some participants had participated in the online interviews, met the students or I during field work or both. The workshops had participants from HISP Malawi, HISP Uganda and HISP Mozambique.

The workshops were structured around the benefits, the challenges and the solutions for user involvement when developing DHIS2.

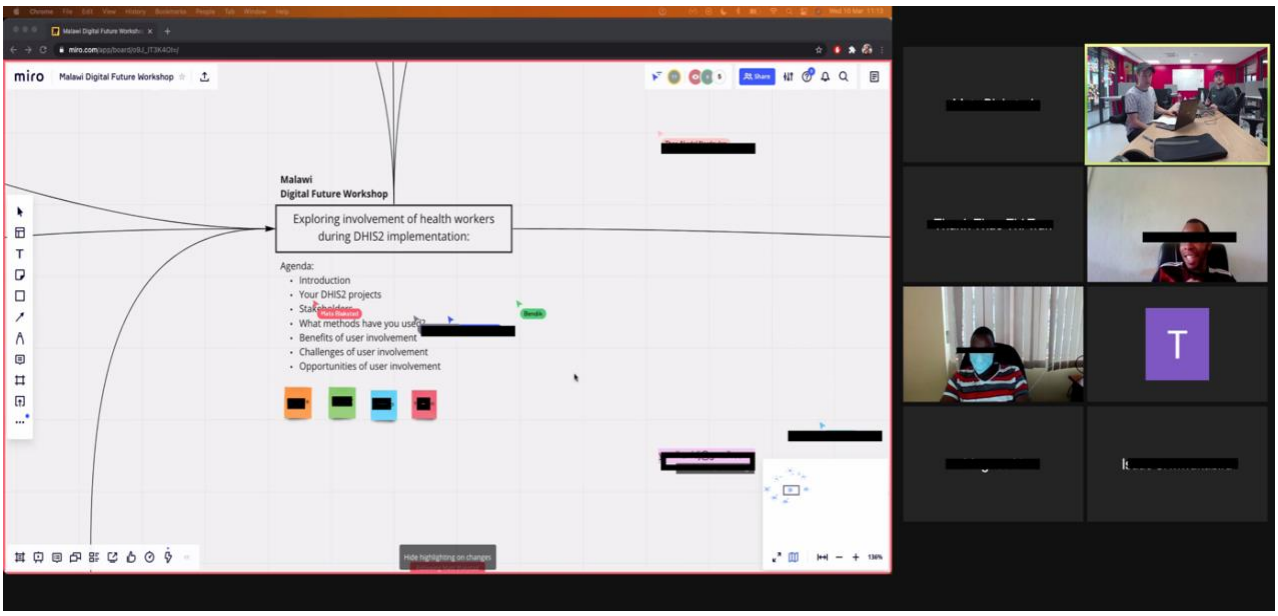


Figure 13: Virtual workshop

The workshops were useful for me to "informally" evaluate my findings done throughout the work of this thesis. Many of the findings done throughout the study were mirrored during the workshops, and I gained valuable insight into where the project could go next.

Documentation of the research process

During all my empirical work, I made sure to keep my work documented. While being in the field (Mozambique and Malawi) I noted down quotes of interest, interesting observations around the practices of the HISP team and other points I wanted to explore further in field notes so that I could analyze and later formalize this knowledge. This is an important aspect of anthropological work, as making sense of "chaotic" social situations were key, these notes and observations were often joined by photos to add further depth to the notes. Often at the end of the day, I would read through my notes and construct this into more coherent text, allowing me to return to these data now a year later. This initial scribbling down of my impressions and notes served as the first informal form of analysis, as I already then started to see patterns and search for these patterns in my future empirical work.

In my entire research process, I also kept a study diary, noting down in a "living document" all the activities and decisions I made throughout my study. This was an important aspect of my study so that I could follow my own analysis, decisions making and general thoughts and activities I partook in. The living diary allowed me to see why I made a decision at one point in time, making the writing of this thesis in a retroactively way possible. This document also contained questions for further explorations, new ideas and plans for future activities.

4.4 Analysis of data from the diagnostic data collection

Methods for analysis

As stated over, the analysis of my data happened continuously throughout my study, which is very much in line with Crang and Cook (2007) as they argue that data gathering and analysis should happen in tandem. This, along with the knowledge already present when going into the data gathering informed my understanding of the themes and patterns I saw, much like a lens. These concepts are for example user centered design, to give a name to the methods and practices I understood or implementation-level design to understand what, how and why the HISP groups work the way they do.

The process of data gathering, design work and evaluations all happened in tandem in the Fall of 2020, making the process much like a "see-move-see" pattern as explained by Bakke and Bratteteig (2015). Here the designer "sees" a situation, a diagnostic finding or design feature. "Moves" based on what they see, this could be deciding to figure out something new based on analysis, design a new feature based on feedback, and then sees again, evaluated the outcome of the "moving" (Bakke & Bratteteig, 2015).

4.4.1 Thematic analysis

One of the methods I have utilized the most to see patterns, organize my findings and thus help analyze my data have been thematic analysis(Braun & Clarke, 2012). As a form of analysis, the thematic analysis allows us a as researchers to learn from the data itself. In my case, the thematic

analysis was inductive, as the data gathered and the themes were all derived from the empirical data gathered.

One of the main goals was to understand what happens where, and where the most challenges are present in a typical implementation process, and then further how this could be supported through design. This was done through the development of a process model and the typical activities that takes place in it, as seen below (figure 14).

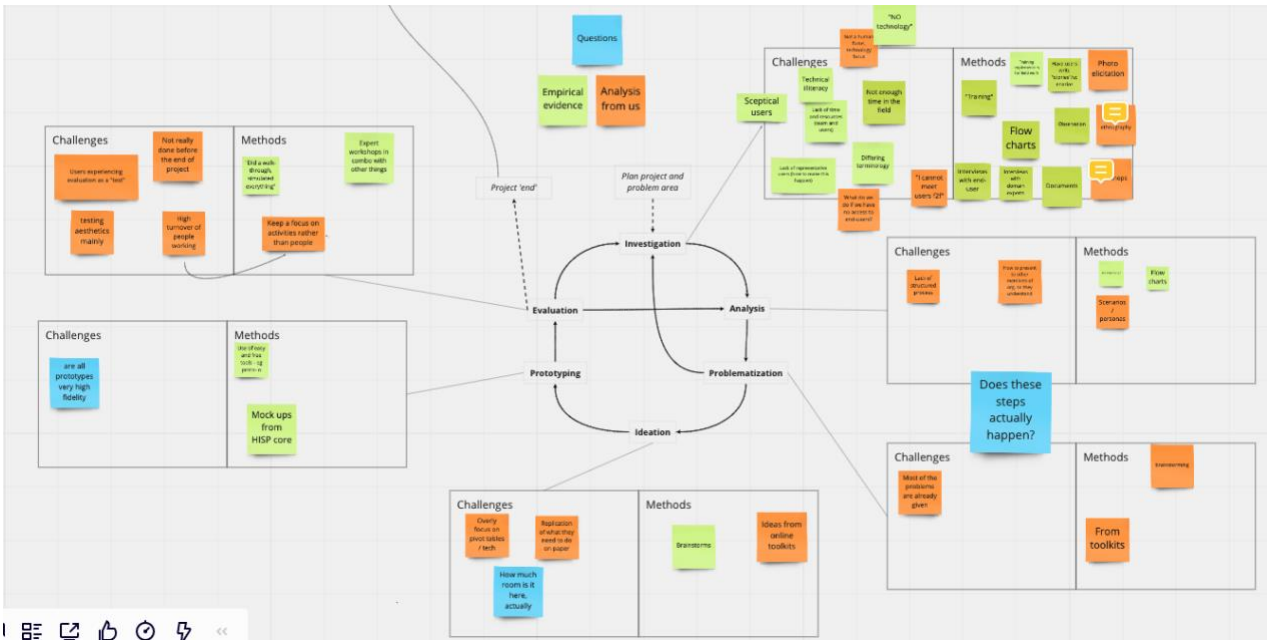


Figure 14: Analysis of challenges centered on a UCD model

For all the activities identified, we (another master student and I), plotted findings from our empirical work along the lines of challenges and methods used. This served to give us an overview and understanding of "what happens where and when", structuring our findings and help establishing patterns, an example is seen in Figure 15.

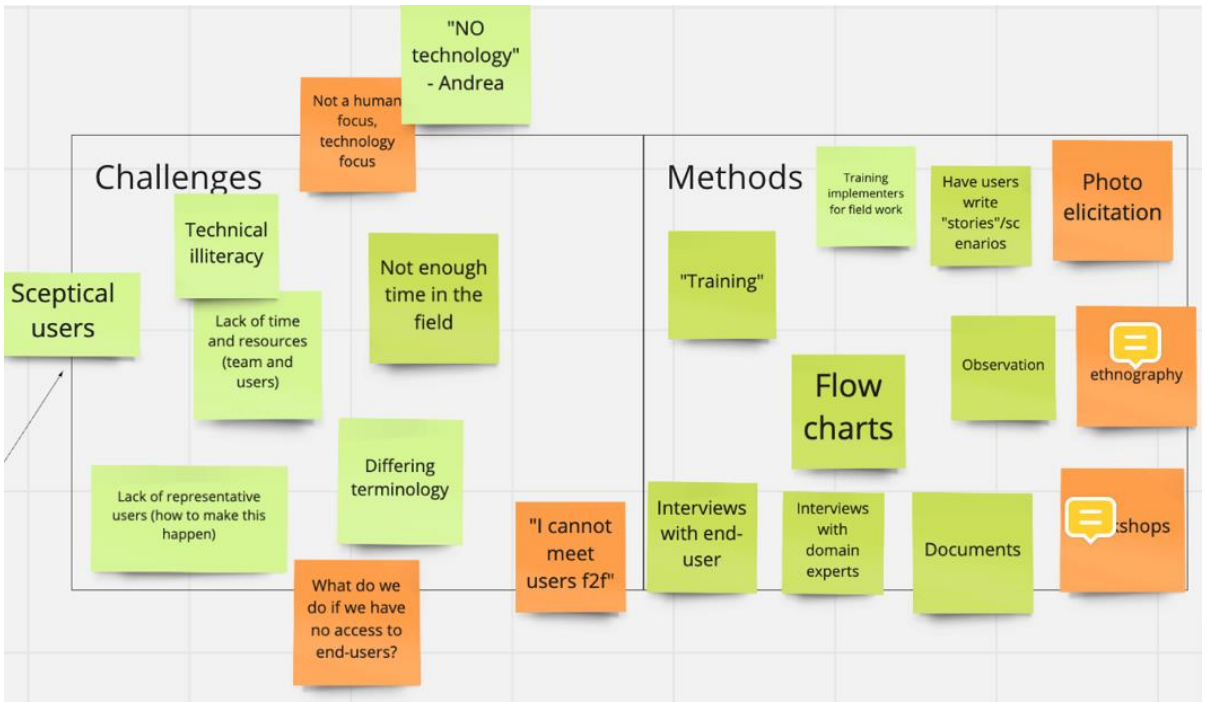


Figure 15: Example of analysis theme, challenges and methods used during insight work

Much of the thematic analysis and co-analysis was done online due to the ongoing covid-19 pandemic. To collaborate with practitioners and other master students the online collaboration software Miro[1] was used.

One of the most important outcome of the thematic analysis was to derive *design requirements* that needed to be met in the final prototype. By structuring the analysis around themes, and especially their challenges several patterns emerged from the data. These common patterns cutting across the empirical data collected allowed us to understand the requirements that a resource like this would need to meet. These requirements were present in the design and evaluation work, and would later become the basis for the design principles presented in this thesis.

4.4.2 Co-analysis

Much of the analysis work that has taken place during this master thesis has been co-analysis, either with participants or with fellow master students.

While conducting field work in Mozambique and Malawi, my fellow design lab members and I would present our findings, analysis and understanding of practices to the DHIS2 practitioners we

[1] https://miro.com/

would be working with. This was so that we could "test" our findings, and therefore make sure we understood the context and practices in accordance with the participants. This "validation" of our findings were important as our cultural context and viewpoints are quite removed from the context which we are studying.

With my fellow master students in Oslo, I have also co-analyzed to help make sense of the data gathered and to see patterns and themes. The co-analysis was done through asking questions of the data, like "why do we think the practitioners mean by not having access?" or "what are challenges that persist across interviews and workshops". Many of these questions helped giving me a deeper understanding of the thematic analysis as well as gaining multiple inputs due to the different experiences and understandings we all had of the data presented.

4.4.3 Design Principle Abstraction

The design principles presented in this thesis are heavily empirically based. They have been derived from the final prototype after its evaluation in a reflective approach of design principle establishment (Möller et al., 2020). As design principles are supposed to be a prescriptive from of knowledge, to help guide other researchers or practitioners, I had one central question in mind when deriving design principles throughout the process;

> *"If I would give someone else guidance in what a method toolkit to support UCD methods should be, what are the most important features and aspect to include?"*.

This thus makes the principles closely related to the artefact itself, rather than the process of making it.

In the figure below (Figure 16) I have presented a simplified model of the process of deriving design principles. Diagnostic work and evaluation continuously informed the design process and being informed by it. Existing toolkits informed the design process by giving insight on design features and how challenges can be met. The process itself provided me with a final prototype which was used to present and derive design principles. All the activities helped me gain knowledge and insight which moved towards the final design principles, as these are the culmination of all the activities taking place throughout this thesis.

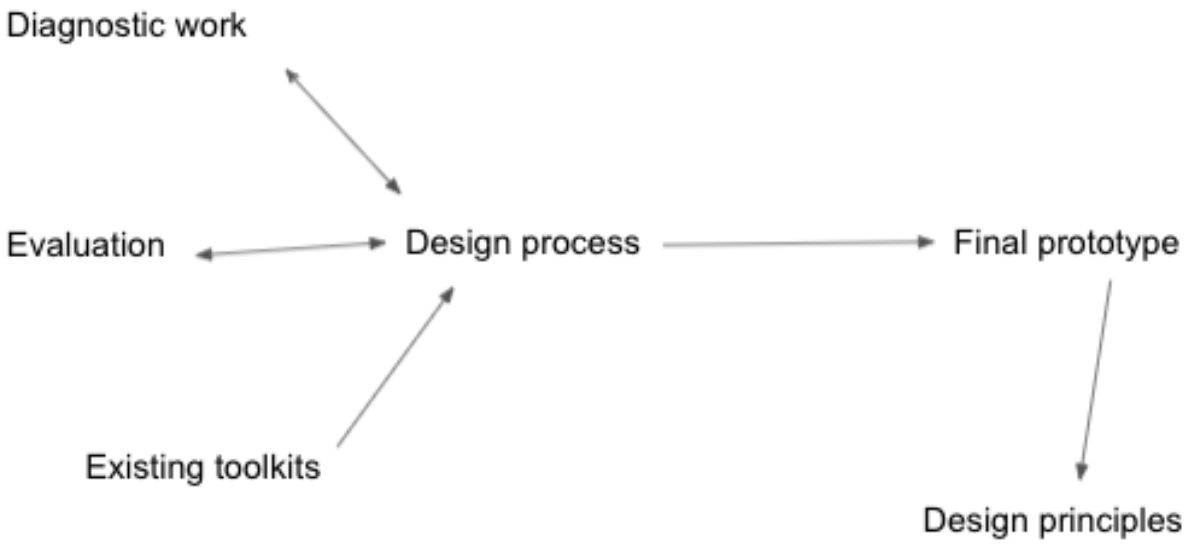


Figure 16: Activities informing design principles

Summary of diagnostic work

In the diagnostic phase, I aimed to understand the practice of the HISP groups related to user centeredness. It was important to apply a plethora of methods as practices are diverse from organization to organization and practitioner to practitioner. The diagnostic work started to identify possible point for interventions to take place, but due to the ongoing pandemic, the focus changed. This made the work move from a primarily physical space to a virtual one, making me change approach. The change from physical to virtual research was a change as physical data collection was something I was trained and comfortable with, while virtual data gathering was a new approach to data collection I learned while doing the work. Learning of the strengths and weaknesses of this type of work as it happened.

4.5 Design work

Design work is one of the most important aspects of a design science research process, often referred in the literature as the "build" phase (Gregor & Hevner, 2013). The data collected during diagnostic work was pivotal in informing the actual design of the resource and its content and form. The experiences of DHIS2 practitioners involved in the diagnostic activities presented over

were the main informative insight that helped the design which were translated into design requirements (presented in 5.5). This was done deliberately as a bottom-up rather than a top-down process, as relevance of methods and challenges as well as familiarity with these was the main goal of the design work.

4.5.1 Design research into online toolkits

Method toolkit have been used by designers in the IT industry to a significant extent, supporting processes and methods for design, investigation, evaluation, ideation and more. As a part of the design work, I did a structured review of method toolkits and toolboxes for user centered design online. I reviewed popular toolkits in terms of content and form, to see if I could use this insight in my own design, for our own domain. It is worth noting that most of the design toolkits online are highly generic and unspecified for domains.

Below (table 6) are the criteria we decided to review the toolkits found online along to make sure the review was a structured as possible. In this manner, it was easier to compare and contrast the toolkits to each other.

Site structure	How do the toolkits present a design process?
Techniques	What techniques are used where?
Filtering	What do the different toolkits filter on?

Table 6: Review criteria, existing toolkits

Researching toolkits was a fruitful design activity, as I gained insight into innovative ways of presenting my findings and design features that could meet the design requirements I had identified in my diagnostic work. Specific design solutions such as presenting outcomes, what to pair techniques and methods with as well as expected outcomes and quick stats like time and materials needed were all features that are present in my design too as these were aspects I wanted to explore specifically for my case.

I also compared the toolkits with the design requirements identified from diagnostic work, to evaluate whether or not these were met and if so, to what extent. This helped me understand what was needed by a toolkit if I were to develop one, as well as combine features that successfully met

design requirements, but were not in the same toolkit. For example, how prototype 1 could meet design requirement 1, and prototype 2 could meet design requirement 2 but none met both requirement 1 *and* 2.

4.5.2 Prototyping

"The goal of prototyping isn't to finish. It is to learn about the strengths and weaknesses of the idea and to identify new directions that further prototypes might take"(Brown, 2008, p. 3). The design of the final prototype went through four main iterations, the first being an initial exploratory iteration, an iteration focusing on system architecture, an iteration for exploration of interaction and flow and the final iteration which had as an aim to develop a prototype for evaluation, which was influenced of the other iterations as well as theory and empirical insight. The "final prototype" used in the evaluations (see 4.6), was used to evaluate the concept, rather than finish the design process. With each iteration, the fidelity of the prototype would get higher, allowing for a more precise feedback to take place.

Below are summaries of what design activities and rationales that took place in each of the four main iterations of the artefact. The findings from these, alongside with design examples are further discussed and presented in Chapter 6.

Iteration 1

The design of the artefact started with low fidelity sketches designed in PowerPoint to explore potential structure and what is needed from the logical structure of the site. Much of this iteration happened in tandem with initial analysis of the insight gathered from the field study in Mozambique and Malawi.

Iteration 2

The prototyping and evaluation was done with another student who also was a developer and an expert in developing web applications like the toolkit was intended to be. He had experience working with DHIS2 previously and was a crucial part of the design of the internal structure of the toolkit itself. Feedback on the structure of the system was crucial as the expert pointed out the feasibility for maintenance and flexibility of the prototype.

Iteration 3

Iteration 3 was focused on flow inside the system and exploration of aesthetic properties like presenting information of the prototype where I moved on from using power point to Adobe XD as a prototyping tool, which allowed me to explore more properties. During this iteration, many members of the Design Lab were involved in providing me feedback on the design as many of the members at this point had gained insight into the DHIS2 ecosystem from their own master theses.

Iteration 4

The final iteration had as a goal to provide a higher fidelity prototype that could be used to evaluate the utility of the resource as well as the design requirements found from the diagnostic work. For the prototyping, I used Figma[2], a prototyping tool that allows for easy collaboration both internally (between me and other students) and externally (between me and DHIS2 practitioners). This allowed other students to give me feedback on the design when we all were physically removed and working remotely, as well as provide the DHIS2 practitioners I wanted to evaluate with an opportunity to not just see static screenshots and ideas but be able to interact with the prototype itself. Which was one of the most useful aspects of this prototyping.

4.6 Evaluation

One of the most central activities in a design science research study is the evaluation of artefacts (Venable et al., 2016). The evaluation of artefacts in DSR is important twofold; first due to the establishment of utility (practical problem) and the evaluation either explicit or implicit of design principles (theoretical). Drawing to Venable et al. (2016) I discuss my evaluation among two questions, when to evaluate and why to evaluate. The first is concerned about timing of evaluation, *ex ante* or *ex post* before or after implementation of artefact (Venable et al., 2016). As the authors (Ibid.) notes, these types of evaluations exist on a scale, the two terms occupying each extreme. However, drawing on the definition as presented by Venable et al. , ex ante evaluation is the most appropriate type of evaluation taken place in my study: “the predictive evaluation which is performed in order to estimate and evaluate the impact of future situations”(Venable et al., 2016,

[2] https://www.figma.com/

p. 79). The second question is concerned with the question of whether the evaluation is formative or summative. This establish what the purpose is, and what you aim to do with the feedback you get. My study was, as elaborated under, mainly concerned with formative evaluations, as the primary goals was to establish the utility of the concept, for other members of the DHIS2 Design Lab to keep iteratively researching, refining and working on the project.

I wanted to try to establish the utility of the artefact, and as Gregor and Hever notes, *"with very novel artefacts, a "proof-of-concept" may be sufficient*"(Gregor & Hevner, 2013, p. 351). The aim of my evaluations was thus to explore the utility and relevance of the artefact I developed, a method toolkit. Both evaluations were formative, as the method toolkit research project is planned to be continued by other master students in the DHIS2 Design Lab. The goal was thus to evaluate the concept and its utility for other master students to iterate on the concept, its form and function.

As well as the establishment of utility the evaluations also aimed to evaluate the design principles. Though not explicitly evaluated, the principles were reflected and derived from the artefact designed. By evaluating the design features which reflect the design principles, I argue I "meta-evaluate" the principles.

It is important to note, drawing on the arguments of Goldkuhl and Karlsson (2020), that demonstrating utility of knowledge artefacts like methods, techniques and tools to support these holds specific challenges, as all projects are unique. Their performance cannot be measured by metrics and experiments, as the team utilizing these types of artefacts will change from project to project, either by composition, experience or both (Goldkuhl and Karlsson 2020). This therefore asks for a more holistic and cumulative approach to the evaluation of these types of knowledge artefacts, like walk-throughs, demonstrations and the presence in multiple projects.

4.6.1 Demonstration

Demonstration was a major way I evaluated my artefact. I demonstrated the use of the artefact, the contents and the logic behind the prototype. This was done through sending an online link of my prototype to the participant, then having the participant share the screen as we have a walkthrough of the prototype. Throughout the evaluations I continuously asked for feedback on the prototype, trying to establish the utility of the prototype. The participant in the evaluations was a lead

implementer in HISP Mozambique with a deep insight into different projects and an interest in end-users and methods to work with them.

I would ask questions like “could you see yourself using it?”, “who could you see using it?”, “in what types of projects or situations would you use this tool?”, and more specific questions to the different parts of the prototype like techniques or processes. Between the evaluation sessions I would also let the participant keep the link and go back and investigate the details of the prototyped as well as demonstrate it to their colleagues.

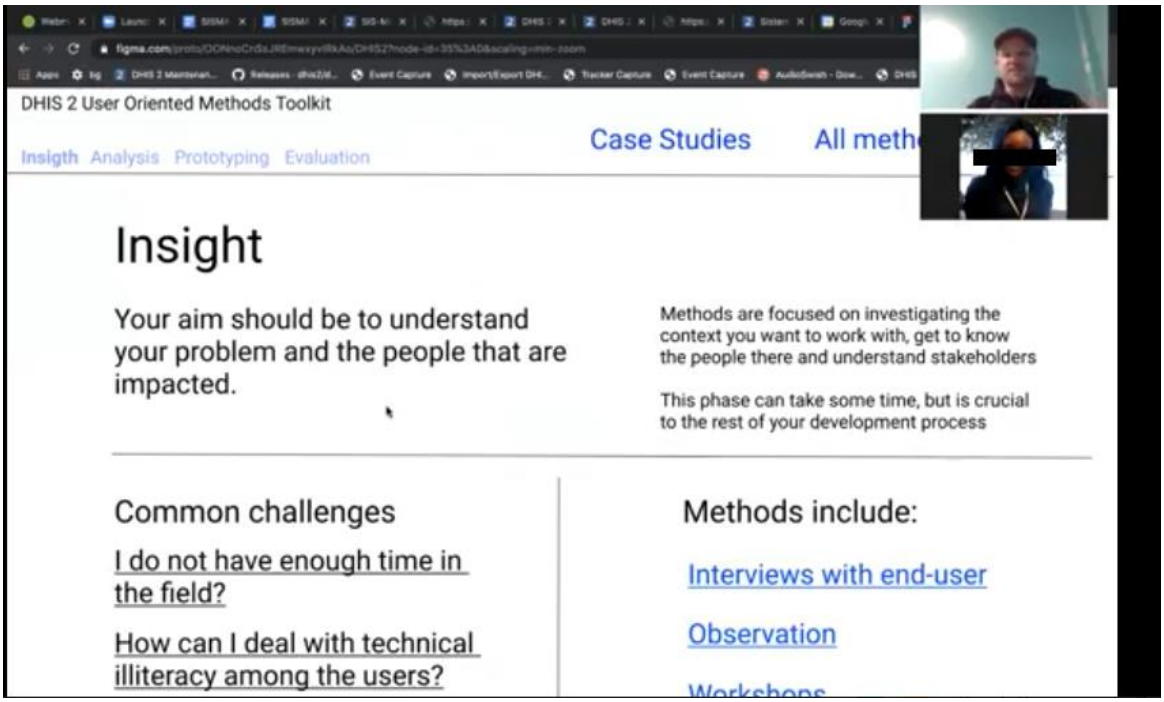


Figure 17: Virtual artefact evaluation

Overall, the evaluations were successful, as the aim was to establish conceptual utility, as well as gain feedback on the prototype to keep working iteratively on it. A learning was that, as I was mainly interested in the conceptual evaluation, I could have been more efficient to send the prototype link in advance, so that the participant had time to investigate the prototype before we met. This was mainly so that they were not too focused on the specific details of the prototype like text and formulations (as this was not the aim of the evaluation).

4.7 Ethical Considerations

Consent for recording and documentation (such as photos) during the virtual interviews and workshops were given by all the participants, where I sent the consent form in advance so that the participants could make an informed decision regarding their participation. This was an important step, as much of the information given and discussions we had were directly tied to their work place, current and past projects as well as co-workers and clients. Some of this information could potentially have adverse effects for the participants, therefore it was important to anonymize my sources so they would not face repercussions in current or future work situations.

During the field work (4.3.2) I did not have written consent from the participants, as these situations were more ad hoc and situational. I did however introduce myself, my interests and purpose when meeting different actors so that they knew who I was and why I was there and whenever I documented personal information like pictures or recorded video I would make sure that all the persons present were asked if I could document and whether or not I could use the pictures taken in this thesis.

The participants of this thesis were not given any form of payment for their participation, but rather did so to help me gain insight and knowledge about them and their practices. Therefore, I found it important to "give something back" as they freely gave me so much of their time. This took different forms, for example during the field visit my fellow master students and myself would present our findings to practitioners. This also went for me producing a report from my visit to Quelimane during my contextual inquiry. Throughout the process that has been a nice way to keep in touch with practitioners throughout the process and from my experience, people enjoy it when people from the outside of the group points out aspects with their practices or traditions. In the HISP network for example, DHIS2 practitioners included in this study had a keen interest in what other practitioners in other organizations were doing.

4.8 Summary Methodology

The chosen methodology of this thesis is Design Science Research, were I have researched, developed and evaluated a method toolkit focused on the challenges and solutions DHIS2 practitioners have in the

utilization of user centered methods when implementing DHIS2. My methodology had three main activities for gaining empirical insight: diagnostic work, design work and evaluation. Although presented chronologically and thematically, they all happened in tandem, one finding in one activity informing the next (described as “move-see-move”.

The following two chapters will present the findings from my methodology, the first (Chapter 5) being the findings from my diagnostic work and the second (Chapter 6) being the findings from the design and evaluation work.

Chapter 5: Diagnostic Findings

5.1 Introduction:

To understand how to support and promote user centered methods in a software ecosystem, it was crucial to understand what current practices are related to implementation-level design. Specifically, empirical insight about the challenges and the solutions to these challenges was of interest in this phase.

This chapter has three main parts, which in turn will be the basis for my design requirements needed to be met in the toolkit:

1. Implementation process
2. How and when are users involved in this process?
3. What are the challenges in involving users in this process?

These three areas have been the focus of my diagnostic work, and the insight of these have been used to derive design requirements from to be met by the designed artefact and in turn design principles.

5.2 The implementation process

In the next section, I will present findings of how a typical DHIS2 implementation process happens in the organizations I have investigated. The process of implementation and configuration presented here from the empirical data was crucial to understand so that the artefact designed could have the best possible fit in the context of use. It was crucial to understand this process on the practitioners' terms so that I could integrate this into the design in the most successful way.

It is important to note that the presented findings are a synthesis of the findings from multiple organizations across multiple projects and countries, but some similarities have emerged across. I will also present some examples of projects to show the diversity of projects that exists.

5.2.1 Negotiation of project scope

During all empirical work done concerning implementation process, the negotiation of scope has been a central part of the beginning of every project. This negotiation between the client and the developing team (HISP group / DHIS2 practitioners) is what sets the scene for the project. The flexibility for design is established during this phase, what a developer in HISP Malawi dubbed *"design governance".* The amount of flexibility in this governance is intimately connected to the requirements given by the client organization.

The negotiation starts with the submission of a proposal from the implementing organization, where the general process and methodology to tackle the problem is presented. This is often used to determine the timeline of the project and accordingly the cost. In this phase, the HISP group submitting have an opportunity to argue for the use of a user-centered methodology, as reported in Malawi and Mozambique. Continually during out conversations, the team in Mozambique would state that they need to "sell" the idea of UCD when proposing projects to underline the importance of this approach.

5.2.2 Requirement gathering and refinement

After the scope of the project have been negotiated with the user organization, a process of requirement gathering and/or refinement usually starts. A major activity during this phase is what both HISP Malawi and Mozambique calls the "situational analysis".

Figure 18: Requirement gathering

The situational analysis is where the team concerned with a project gains contextual information central to the development. The aim here is to understand the context for the software, including the users' needs and the context in which these exists. As one senior member noted*: "If you don't do the situational analysis, you do into the place blind*". At the end of this activity the team produces an *inception report,* an important artefact for the development of requirements.

It was important for the members of the team responsible for implementation that this phase should be explicitly stated to happen in the project contracts due to the importance of this activity; *"To*

provide a useful tool, there needs to be a contextual understanding" (senior HISP Mozambique member). The members of the team also noted that ideally this should be done at the very beginning of the project, but it takes a significant amount of time, underlining the importance of the scope and mandate of the project.

Following the situational analysis and the inception report, the requirements given by the client is refined. This is through a process of negotiation, where the HISP team presents of recalibrate the requirements based on new insights. The HISP team noted that there sometimes are conflicts between the HISP team and their gathered requirements and the clients. One HISP team member noted, *"we are often fighting a battle for the users".*

Some HISP members also noted that they prefer to work with the *"specific requirements provided by the client*" rather than specifying and refining their own. As the requirement gathering process is timely and resource costly for the team.

5.2.3 Development

Following the establishment of requirements, the configuration of the software begins. In a "typical" UCD process, there should be some iterative design through prototyping, which was something we were interested in as well and the practice of prototyping had significant change from organization to organization. Usually, the prototyping and mock-ups were dependent on the scale of the project and its adjacency to DHIS2.

Depending on the type of project and team composition, prototyping was done with implementers and developers only, and sometimes with the clients. Prototyping was done depending on fidelity, and the tools used were different accordingly like the use of Excel spreadsheets for prototyping pivot tables. A common theme across HISP groups and DHIS2 practitioners were that prototypes rarely were brought into the field, making an iterative process centered on users difficult due to users not participating in the evaluations and the development team not getting the users feedback. Discussions around this emerged, and usually this was due to lack of time to *"prototype changes".* As one developer in HISP Mozambique noted when asked about why he does not prototype changes or sometimes not prototype at all; *"sometimes I feel like it is just a waste of time, I just want to start coding*".

There is also some iterative negotiation present at this stage in the process, what HISP Malawi calls *"inception meeting".* Here the team and the client would get together, and decide on what needs to be done and what to do in the following iteration, establishing both methodological and technological approach. The purpose of these meetings is to create "inception reports", functioning somewhat as a contract of what needs to be done and how it should be done. The development team has some leeway of deciding how the process should be done during these meetings, allowing for advocacy of user centeredness.

5.2.4 Evaluation and user-training

Evaluation of projects and solutions is usually centered on the validity and validation of requirements from the client. This evaluation is often summative in nature, as it aims to satisfy the client and the project requirements. Some HISP groups noted an approach more focused on testing internally within the group during the iterations, through this happens in a more informal manner.

Depending on the project and contract, the implementation group might have the responsibility of end-user training. During this activity, the organization trains the intended users in the software and how it could support their work practice. Depending on project and contract, the implementation group might have a small pilot project before full end-user training to make sure the infrastructural context is supported in the solution. If the implementation group do not have the responsibility for the training, they usually develop and deliver end-user manuals along with other documentation.

5.3 Involvement of users

A common finding across the HISP groups and DHIS2 practitioners was that requirement gathering/insight work and evaluation was the activity when most of the interaction with users took place. This part will present the practices of involvement of users and what part of the process this takes place in and how it is done.

5.3.1 Involvement during requirement gathering

During the gathering of requirement and gaining of insight into the use context, almost all of the HISP groups and DHIS2 practitioners reported some type of user interaction or involvement taking place, though the extent of this was highly varied. The involvement and interaction with users went from living labs in Malawi, observational field trips in Mozambique to interviews, workshops and reports from the field in other HISP groups.

This was further echoed in my participation in a "situational analysis", the contextual inquiry, with the HISP Mozambique team. A wide array of methods were used during the trip, which was mirrored during the online interviews and workshops later in my stud, these methods being interviews, workshops, observations, official meeting, extensive documentation of the use context through photo and video to mention a few.

The HISP groups and DHIS2 practitioners argues that many different methods to understand users was beneficial. As one participant in an online workshop from HISP Malawi noted concerning the use of multiple methods; *"observations are important to get more than verbal agreements".*

5.3.2 Involvement during development.

Throughout the development process, there is rarely users involved in the typical activities. As noted over, prototyping and development is a crucial part of the implementation and configuration process – this is where the local design of software happens. Ideally, many of the HISP groups wanted to have a more iterative approach, which in turn would allow for more user involvement during the "middle" phases of the project but due to tight schedules and time and resource restraints this is often challenging. An implementer from HISP Malawi addressed this *"they [the client] keep adding features and requirements, but the timeline stays the same",* making an iterative process involving users and specific features difficult.

For the projects that had more flexibility and iterations, the HISP groups argued for the possibility to change the methodological course throughout the project, allowing for more user involvement.

5.3.3 Involvement during evaluation and user training

Most of the interaction with users and the tail ends of projects took place during evaluation and specifically user training. As noted over, the iterative nature of projects was varied, and the methods for evaluation and training differed from project to project, but it is rare that evaluations were formative throughout the process. Most of the evaluations taking place in HISP Mozambique are usually done with the heads for clinics, the clients or leaders in the community, rather than the end-users of the software itself. Throughout my empirical work, this emerged as a challenge as many HISP groups and implementation specialists struggle with gaining access to users.

During an online interview with an American consultant firm working with DHIS2 in Malawi, they reported a unique solution to this problem. They often struggled with leaders participating in workshops for evaluating solutions at health clinics, when they wanted to evaluate and work with end-users. To overcome this, the implementation team started calling the evaluation "end-user training" rather than "evaluation workshops" making sure they got access to the end-users they were interested in coming into contact with.

Depending on the contract, the HISP groups are responsible for end-user training. This is where most of the usability issues emerge, since they are not just evaluating with senior members or leaders. As one implementer noted, *"the real problems you find there when you are working with the user".* Many of the HISP groups gained valuable insight in the user trainings (see Figure 19), especially concerned with usability issues and contextual factors, but often there was little possibility to *"do anything with it"* as one implementer noted. This is usually due to the contractual issues, where the project ends either with user-training or a user manual. One of the senior members of HISP Mozambique said that, if they find issues, they often produce a report arguing for how they can meet these and ideally prolong the contract.

Figure 19: Notes from user interaction, Mozambique

During one of the online workshops held during Winter 2021, when discussing this issue, one practitioner argued for trying to involve users earlier in the process, ideally on the planning.

5.3.1 Examples of projects with user involvement

The online empirical work shed a light on all the different types of DHIS2 projects the practitioners were participating in, mirroring much of what was found during the field work. Throughout the HISP network, there are a multitude of projects, and all different. As one DHIS2 practitioner noted when presented with my understanding of a processes model during an online interview, "*yes it looks like that sometimes, but it really depends on the project*". Much of the different practices and challenges in a development and implementation process throughout the HISP network stems from the different origins, contracts and types of projects the groups are working with. Below are some

examples of projects to help illustrate the diversity of projects I have encountered during my empirical work.

HISP Mozambique works on a diverse set of projects, and I will use this group as an example of the diversity of projects present, as this is one of the groups I have the most insight into. They have projects more typical digitization projects such as focusing on pivot tables and making written records digital (as seen in Figure 20).

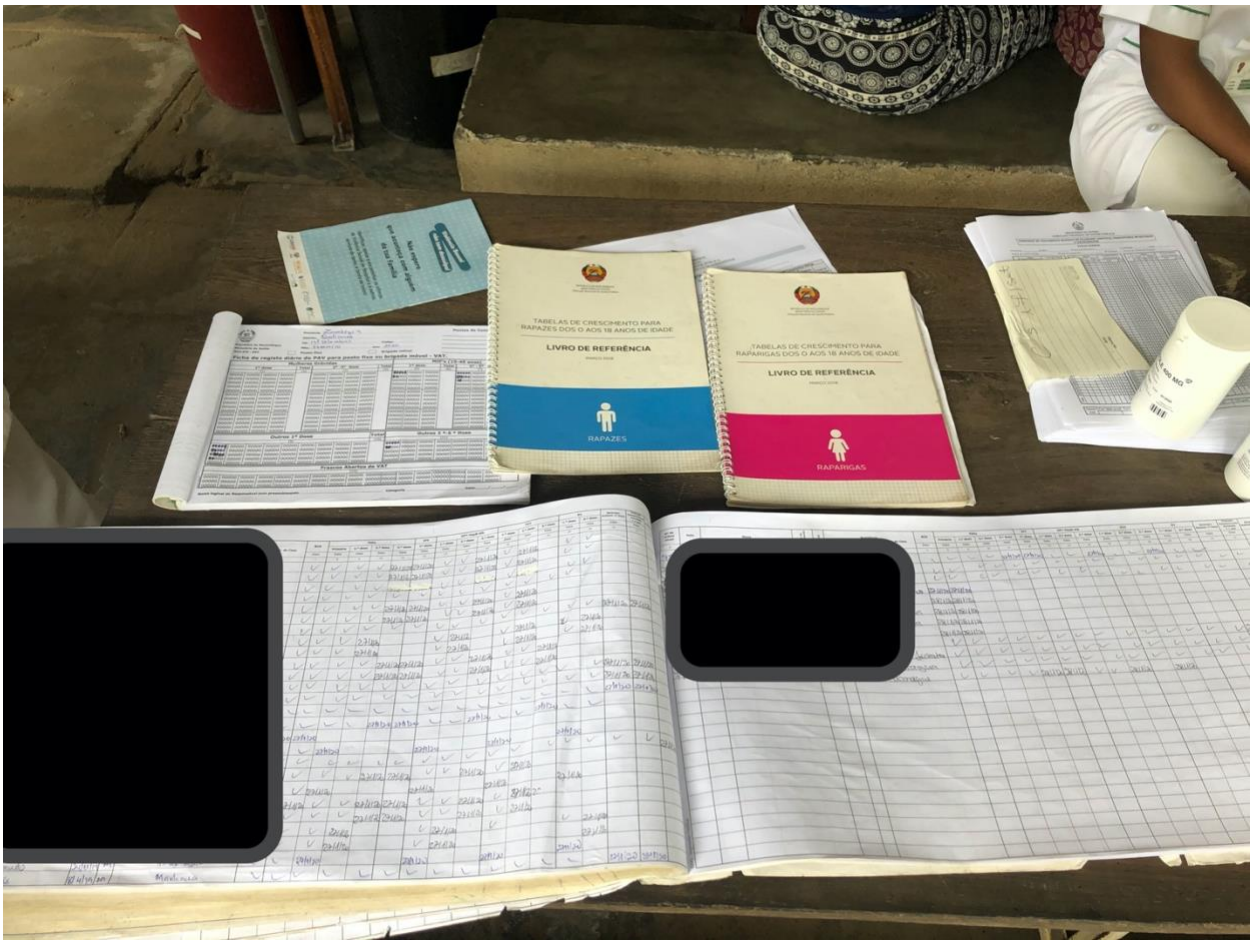


Figure 20: Written records, clinic Mozambique

However, they also focus on innovation and has a portfolio of projects that diverge from the "typical" DHIS2 projects, some of these are further elaborated on under:

- **ePompar** is a community saving application, where community members save money locally for the community outside of a bank. It was initially a digitization effort, but has grown to become more community practice oriented.

- **"Bazar"** is HISP Mozambique effort to develop an e-commerce platform where the aim is to help local farmers sell their produce as well as inform around best practice.

These projects such as ePompar and Bazar are DHIS2 based but centered heavily on users and their needs, having different approaches to user centered methods and design. Many of the other major projects of HISP Mozambique was funded by large, international foundations such as UNICEF, WHO and PLAN. As opposed to some of the innovation projects over, these projects were mainly technology focused to solve a give problem, under are some examples of these types:

- **WASH division** (UNICEF funded), focusing on mapping indicators for water and sanitation such as wells and water pumps.
- **Health community system** in Angola, focused on reporting and communication between different administrative levels (e.g. Community and ministry of health). UNICEF funded.
- PLAN funded project for **sexual health and pregnancy** in women and girls.

The figure below (Figure 21) tries to place some of these projects on a line from digitization to innovation projects, which are further elaborated on under.

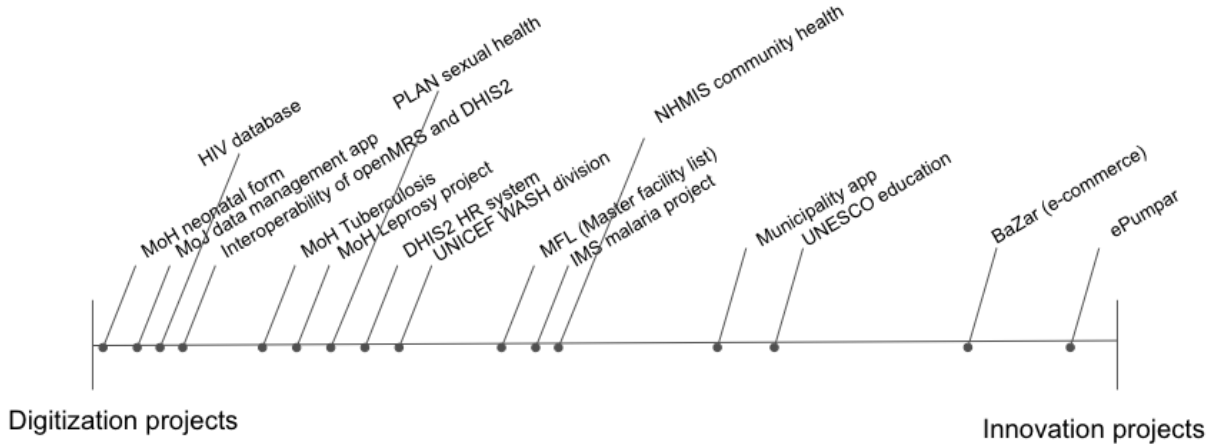


Figure 21: Projects, HISP Mozambique

As some of these examples demonstrate, the projects HISP Mozambique is working with differ in scope, organization and focus (some health, some more broadly community based). The methods used in all these projects were highly different, as some involved active participation from users

(eg. ePompar) and others involved more negotiation with the clients (women and girls sexual health).

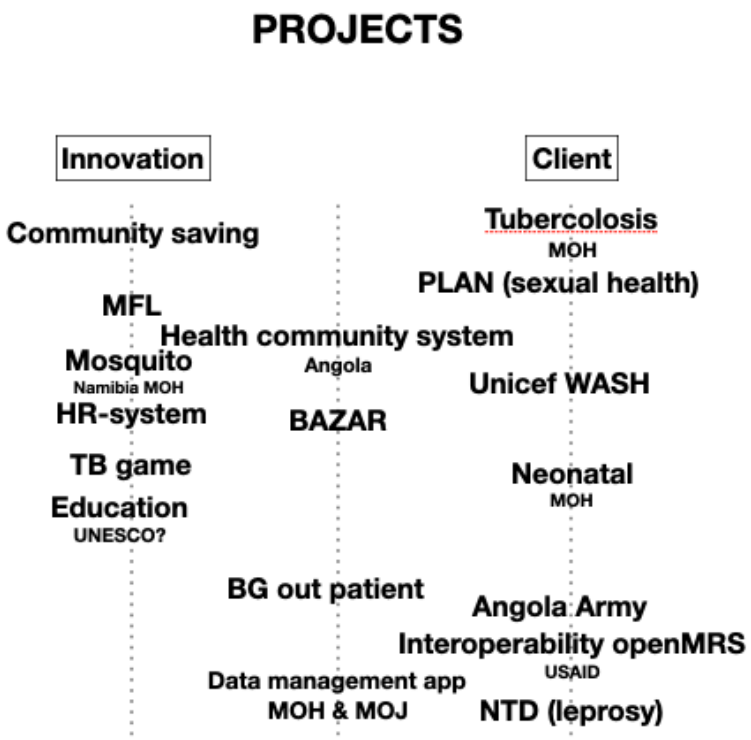


Figure 22: Source of projects HISP Mozambique

The figure (Figure 22) above represents another aspect of the development process in Mozambique, where the projects "emerge" from. Some projects are ordered by a client, like UNICEF, PLAN or the Ministry of Health. These types of projects usually have very set requirements, having established mandates, leaving a design context that is determined by other actors than the HISP team themselves.

Other projects done by HISP Mozambique are, as one senior member put it, *"solutions looking for a problem".* These types of projects are developed by the team, for them to try to sell these solutions to actors like MoH or NGOs. During my field trip to Quelimane, most of the projects investigated were this type, for example the *"community saving app – ePompar".* These types of projects leave the development and implementation team with more leeway in how they want to develop and what processes they aim to follow, especially before they sell the project to a client.

5.4 Challenge of involving users

A central trend was that although projects and teams were diverse and differed highly from each other, similar challenges were present. I have chosen to group these challenges as time and resources, scope/mandate, lack of guidance in design process, technical illiteracy/communication with users and access to users. In collaboration with the previously presented findings, these challenges have been a part of the basis for the design requirements leading into artefact construction and design principle development.

It is important to note that these challenges overlap with each other in terms of source and might emerge in tandem during projects.

5.4.1 Time and resources

A common denominator throughout all the empirical work was the lack of time and resources awarded to user centered methods. Many practitioners belie that user centered methods are very costly, as this is a time-consuming activity, not only due to the budget of the project and tight timelines, but also the added costs of traveling, hosting workshops and being in the field. These logistical challenges to the inclusion of users was often related to the limited time and resources from the project scope. As one of the team members of HISP Malawi noted; *"there is intent, but there are logistical issues that might not match with the process*".

This was echoed during my field trip in Mozambique, where I joined a HISP team in a situational assessment. The team had a very limited time to get useful and important data collection done, during the assessment. The initial resource cost of user involvement might therefore be perceived as more resource draining than *"just start coding",* as a HISP developer noted.

5.4.2 Scope / mandate

Much of the project scope is determined by the client, whether this is the ministry of health or other NGOs, the scope and mandate of the projects determines to what extent a user centered process can be followed. This relates to the lack of resources mentioned over (5.4.1), where the resources allocated to user centered methods and processes might be cut if the scope changes, an implementer in Mozambique noted this; *"..they[client] start cutting, removing activities…".*

The lack of user centered approaches and methods were explicitly noted by HISP Malawi as a problem tied to scope. When discussing this with the team, they noted that the lack of inclusion of these approaches in the calls for tenders being a major hurdle. By not including user centered approaches in the calls for tender, the client either sees no use in it or do not know of the benefits of explicitly involving users in the software process. These reasons make it difficult for the HISP group to argue for the added time and effort it will be to go into the field and work directly with users themselves.

5.4.3 Access to users

Access to users, especially representative users was noted by all the DHIS2 practitioners involved in this study as a reoccurring challenge. Often, when the groups would visit clinics or other health facilitates, they were only given access to supervisors or leaders, not the end-users. This was echoed in my trip to the field in Mozambique, where the team and I had to go through bureaucratic hurdles and meeting with officials to be allowed into the field, then often with supervisors' present at all time. As noted in an online workshop with implementers in Malawi, it was specifically hard to get access to representative users.

In Mozambique, challenges with working iteratively and having access to users came up during discussions with the team. Specifically, this was concerned with often a high turnover of users at health facilities, making the same users follow a process over time difficult. Another challenge noted by the HISP Mozambique team was knowing who and what users' opinions to include, as one implementer said, *"there needs to be enough users with a problem for us to do something with it*".

5.4.4 Technical illiteracy / User communication

Another challenge that was reoccurring in all interviews, workshops and other empirical work was challenges working with users concerning the communication with users. One implementer from HISP Uganda said that is challenging to talk to users like health worker because *"we do not speak the same language",* meaning that the HISP team and the end-users do not really understand each other when working with technical solutions. This difference in terminology was a returning issue,

leading to many of the HISP groups preferring to interact with program directors and managers instead of the end-users as one HISP implementer said, *"it is easier to talk to them [directors and supervisors]",* while at the same time acknowledging this not being ideal. Another challenge an implementer from HISP Uganda noted, was that user often have preconceived notions and expectations, often difficult to meet.

An issue that was noted more places too, like Mozambique and Kenya was that users also was skeptical to new technology at their workplace, making it more challenging to work locally. It was reported that this might stem from a fear of the users to lose their work or have it changed, leading to what an implementer called *"lack of honesty from users".*

Technical illiteracy, the lack or almost lack of knowledge in the handling of digital tools or computer programs, was a source of frustration when working with users for many HISP groups. This is central to the two aforementioned challenged over, as the lack of technical skills and understanding for users makes it hard to communicate with end-users concerning technological solutions and makes users highly skeptical to these solutions.

This challenge might stem from a lack of methods or understanding how to change and adapt methods to "fit" this type of challenge – underlining the need to build capacity and make user centered methods relevant for those who aim to use these.

5.4.5 Lack of design process guidance

During out conversations, the HISP Malawi team noted that there is a lack of guidance in design processes, adding additional pressure of the development team to use available resources in an efficient manner. Some of the team members further said that they feel they do not have the necessarily insight into design processes and approaches they need to execute these in a satisfactory manner or argue for the use of them.

Another factor adding to this challenge is that some projects have a chaotic organization, as client organizations shift or new projects gets added and removed from the organizational portfolio. This reduces the design space for the team, making it challenging to work with design processes. As one DHIS2 practitioner noted; *"Often you feel like you are going in circles, the same objectives but new times and new language".*

When discussing methodological challenges with DHIS2 practitioners that mainly have worked in Malawi, they said that some of the often promoted user-centered methods did not "fit" in their local context. This might be due to lack of access or understanding of users as mentioned over. Often due to these challenges, the implementation specialist interviewed from Kenya noted that the design work is reduced to UI work only, with focus only on colors and shapes, not work flows.

In some of the online workshops conducted as well as interviews, many of the challenges the participants presented were also heavily technology centered, where users were almost seen as a hurdle to conducting the design they wanted. This might be due to a lack of understanding of the general aims of doing user-centered design and methods among practitioners, as the focus ideally should be on understanding users on their terms. Further, it might be reflective of how some user centered methods do not necessarily "fit" well in the contexts the DHIS2 practitioners aim to use them in.

5.5 Summary and design requirements

Unique solutions to issues related to user-centered methods were also reported. As reported from Malawi they had problems gaining access to end-users, as supervisors and senior members would participate in workshops and interviews, not the end-users (data clerks) they were interested in. The team decided to name the workshops "end-user training", rather than workshops, making many of the supervisors and leaders that would participate not come. They used other tactics too, like always having workshops and meetings in the afternoon after lunch, making sure the users they were interested in could participate.

Challenges concerned user centered methods exists in all parts of a development process, but especially during what we dubbed the "requirement gathering and refinement" phase. I argue that this is mainly due to two reasons:

- This is the phase where most of the interaction with users take place and the need for methods to understand users and their context is highly present here.
- There is little interaction with users in other stages of the process, and the participants our empirical investigation gave other phases little attention compared to the investigation.

Based on the presented findings from this chapter I define four design requirements from the DHIS2 practitioners included in this study which needs to be met by the toolkit developed (Figure 23). These are derived from the data presented and are based on the common challenges and findings from across DHIS2 practitioners.

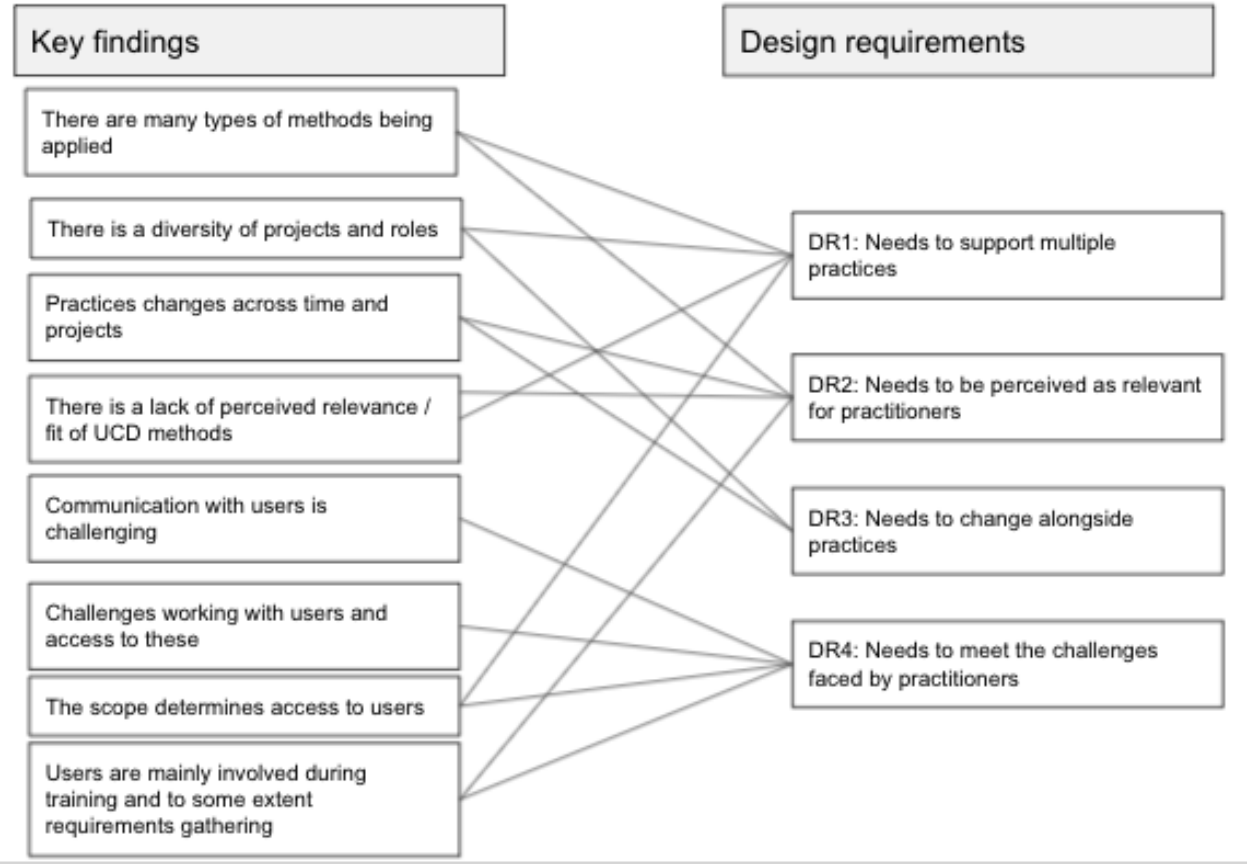


Figure 23: Key findings and design requirements

Chapter 6: Findings Design and Evaluation

This chapter presents findings from the iterative prototyping and evaluation of the *DHIS2 Design Method Toolkit*, the artefact designed in this thesis. Like in the diagnostic findings I will present design requirements derived from the diagnostic findings presented over, as well as how they have been realized through design features in the final prototype.

The DHIS2 design method toolkit as presented in this thesis was iteratively designed as a wireframe as a proof-of-concept for evaluation and the focus has been on conceptualize and create an instantiation to evaluate utility(Gregor & Hevner, 2013). The presented findings are centered on how design features and solutions have informed my toolkit as well as which features I have chosen to explore, but also how and if the existing solutions meets the diagnostic design requirements presented in Chapter 6.

The chapter is structured accordingly; first, I will present findings from research of already existing design method toolkits and present design features that influenced my final design, presented in the next chapter (Chapter 7). Then I will present four iterations of design, focusing on the insight leading to the design, the prototype and the learnings from evaluations of the toolkit.

6.1 Findings of design research of online toolkits

Method toolkits have been a staple for designers in the IT industry to a significant extent, supporting processes and methods for design, investigation, evaluation, ideation and more. They are often used for inspiration for new types of methods as well as new solutions to meet challenges in different design contexts and offer guidance in how to conduct techniques like focus groups, interviews and brain storming sessions. Being a novice in toolkit design, I wanted to explore existing method toolkits to understand how these have been designed by other designers and if there are useful design features that I could use as inspiration to meet design requirements.

In this section, I will present findings from existing method toolkits that informed the design of the artifact, as well as design features that was used in the final design.

6.1.1 Design Requirement 1: The toolkit needs to support multiple practices

There are many types of methods presented in the online toolkits, from familiar methods like interviews to more experimental methods like those presented in Figure 24 and Figure 25. A diversity of methods is something that is well represented in current online toolkits, allowing users to quickly find a myriad of methods. As seen in the diagnostic findings, this is something that is reflected in the "real life" practices of the people working with DHIS2 as well. Where practices are different from project to project as seen for example in HISP Mozambique (5.3.1) , and often throughout a process as well.

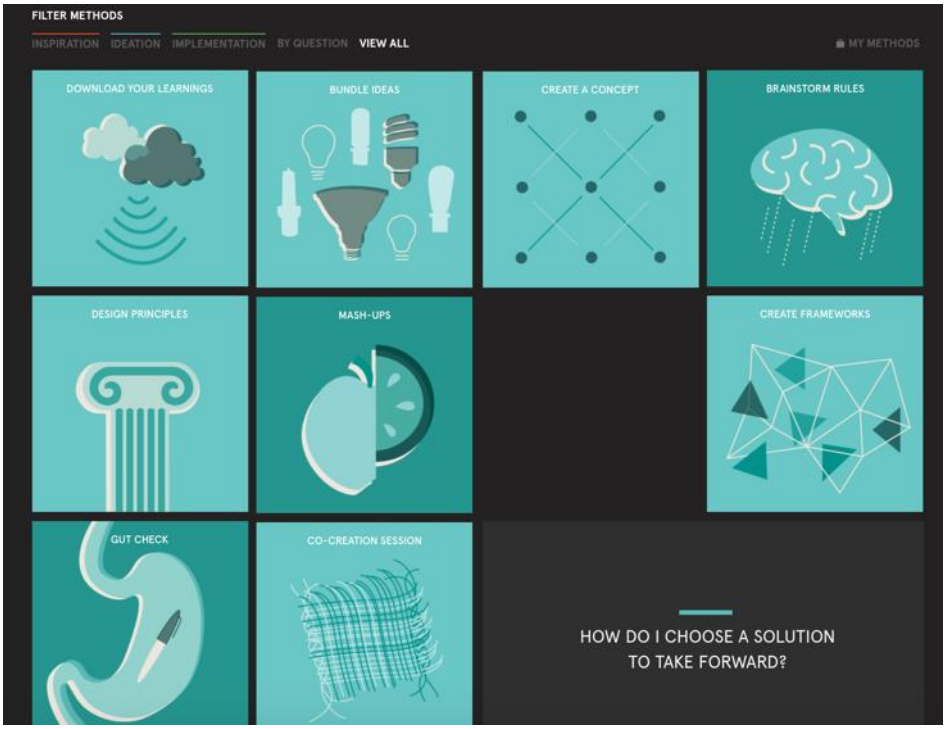


Figure 24: Methods[3]

[3] https://www.designkit.org/methods

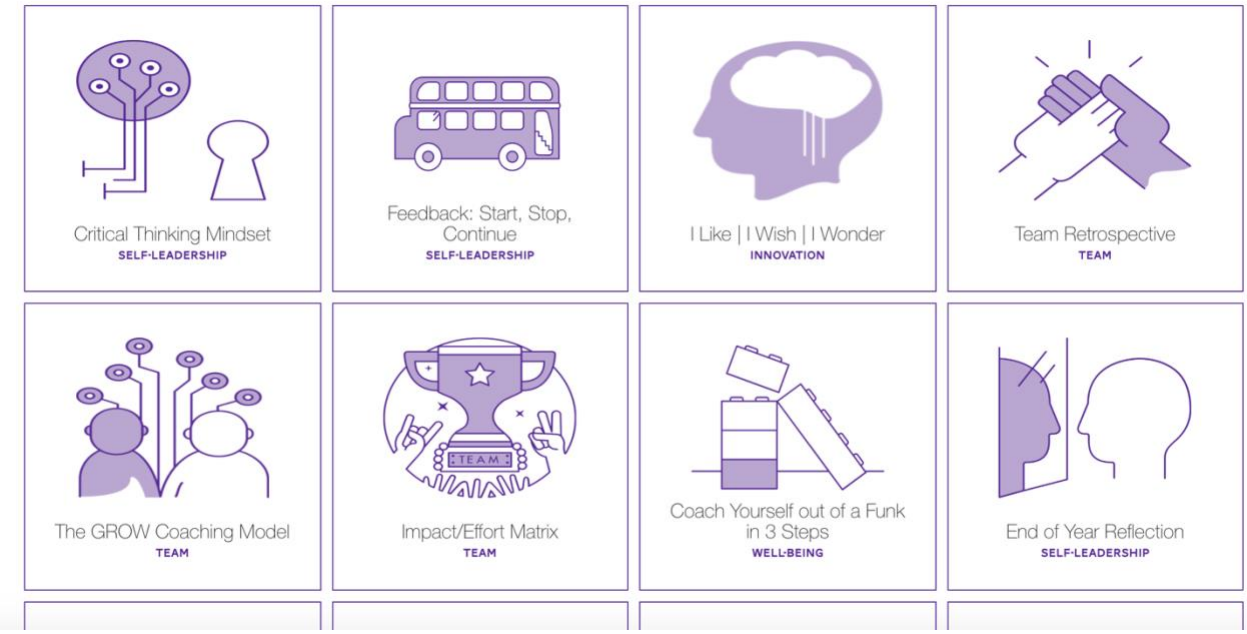


Figure 25: Methods[4]

6.1.2 Design Requirement 2: The toolkit needs to be perceived as relevant for practitioners

The relevance of the toolkit and subsequent content needs to be perceived so by the practitioners that aim to use it. For this relevance, familiarity of accessibility of methods, process and techniques needs to be established. This is done to some extent by existing toolkit like Servicedesigntools.org (Service Design Tools, n.d.), through the establishment of filtering based on your needs, allowing you to get information about methods needed for a specific purpose, as seen in Figure 26. This allows users to tailor a process to your specific needs, which in turn might help support relevance for DHIS2 practitioners.

[4] https://toolbox.hyperisland.com/

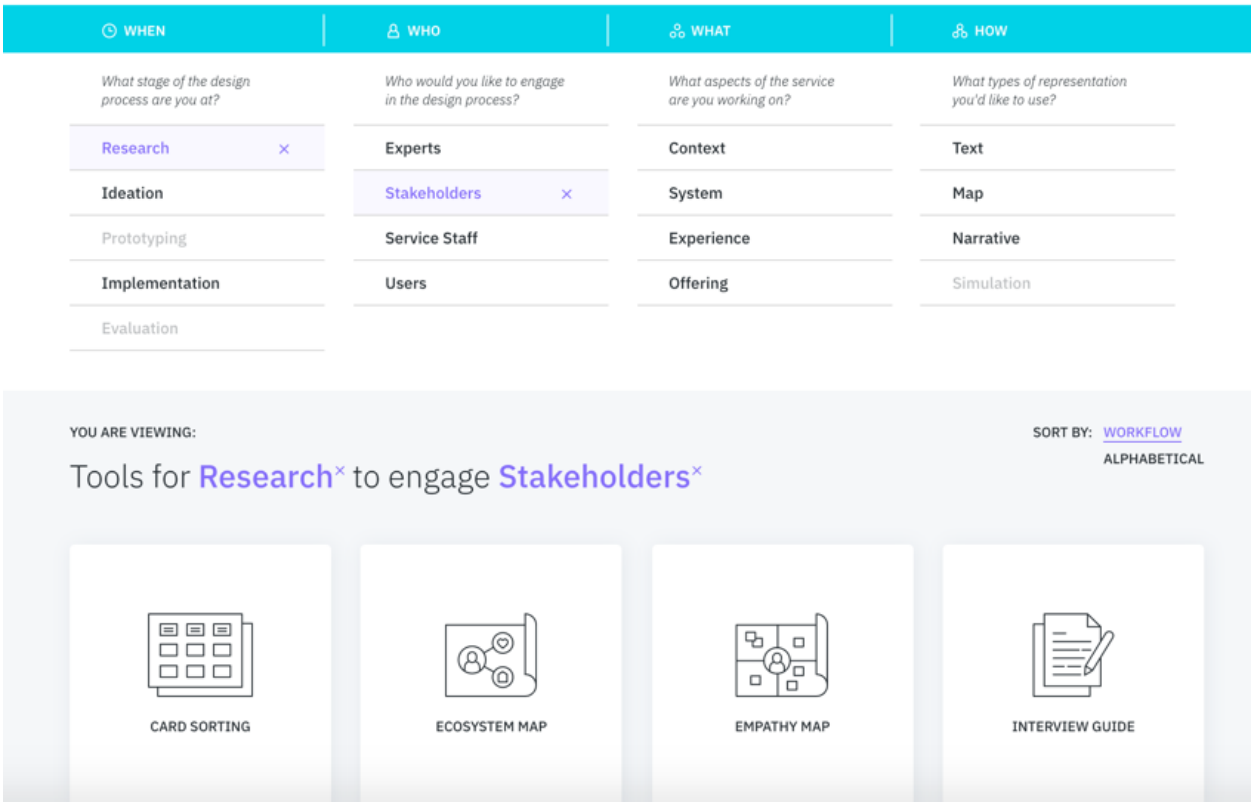


Figure 26: Example of choices to explore specific purpose[5]

Another aspect that I found efficient and adapted from the design research was the presentation of challenges concerning methods. I wanted to explore this aspect further in my prototyping, as I hypothesized that this could be an aspect that successfully could make such a resource seem more relevant. This is because I found a significant amount of challenges with conducting UCD among the DHIS2 practitioners when doing diagnostic work, which were similar across groups and organizations. The presentation of challenges was inspired from the DesignKit (IDEO, n.d.) made by IDEO as seen in Figure 27.

[5] https://servicedesigntools.org/tools

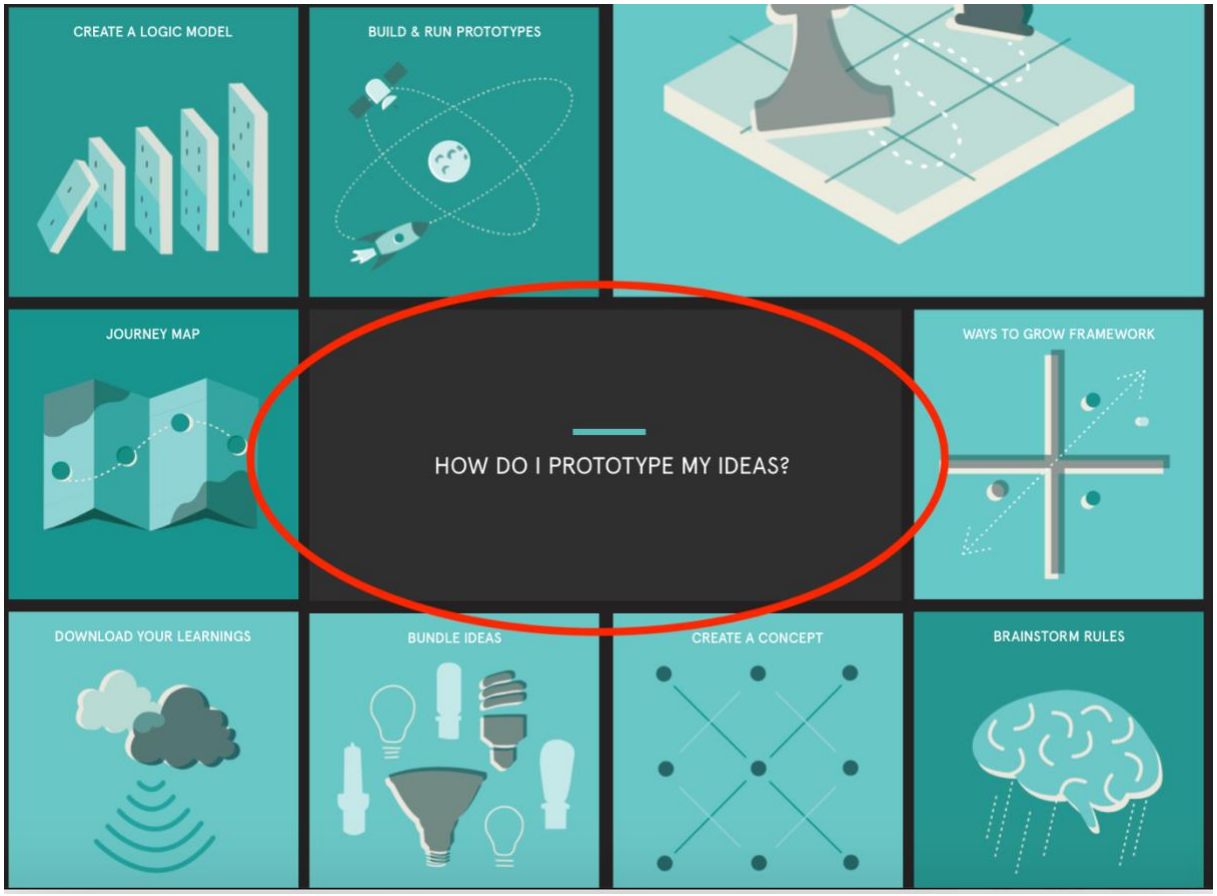


Figure 27: Example of site structure with challenges associated with phases[6]

The challenges in the example (Figure 27) links you to a collection of methods, but does not say anything about the challenge itself, which I argue could make it harder to understand and get the needed support to meet the challenge.

[6] https://www.designkit.org/methods

6.1.3 Design Requirement 3: The toolkit needs to be changeable along with practices

None of the presented toolkits gave the opportunity for input and change from practitioners themselves, and were perceived as top-down from designers experienced in the presented methods. Whether the presented methods and techniques are changed across time is also unclear. Most of the methods presented in the toolkits are also focused bespoke IT projects rather than larger configuration of generic software, as my design requirements are focused on.

6.1.4 Design Requirement 4: The toolkit needs to meet the challenges faced by practitioners.

Many of the toolkit explored had "quick" information like information on how long a method takes (Figure 28) , who is involved and at what skill level the method is, allowing the users to make quick decisions on whether they should or shouldn't conduct the method presented to meet their needs. This could be seen in some of the challenges with the diagnostic activities where the time and effort afforded to UCD methods

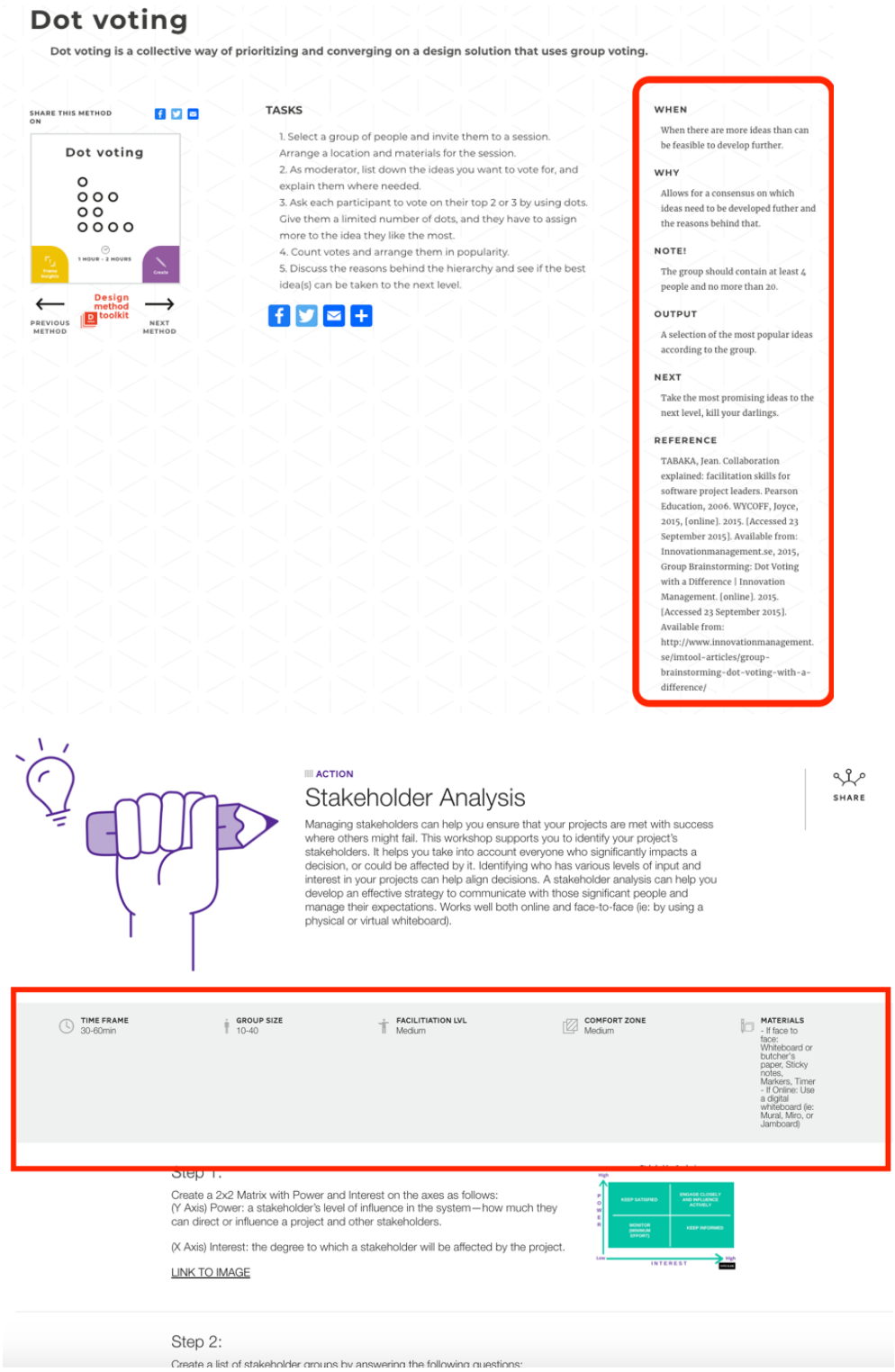


Figure 28: Example of readily available information[7]

[7] https://toolkits.dss.cloud/design/method-card/dot-voting/

[7] https://toolbox.hyperisland.com/stakeholder-analysis

Some of the online toolkits also have presented cases of methods and approaches in "real life" projects as seen in Figure 29. This could help practitioners understand the relevance of the methods presented to a larger extent. Especially with the cases which are similar from projects to projects. Focusing on cases could help practitioners have more tangible examples to relate to, making the methods presented more accessible.

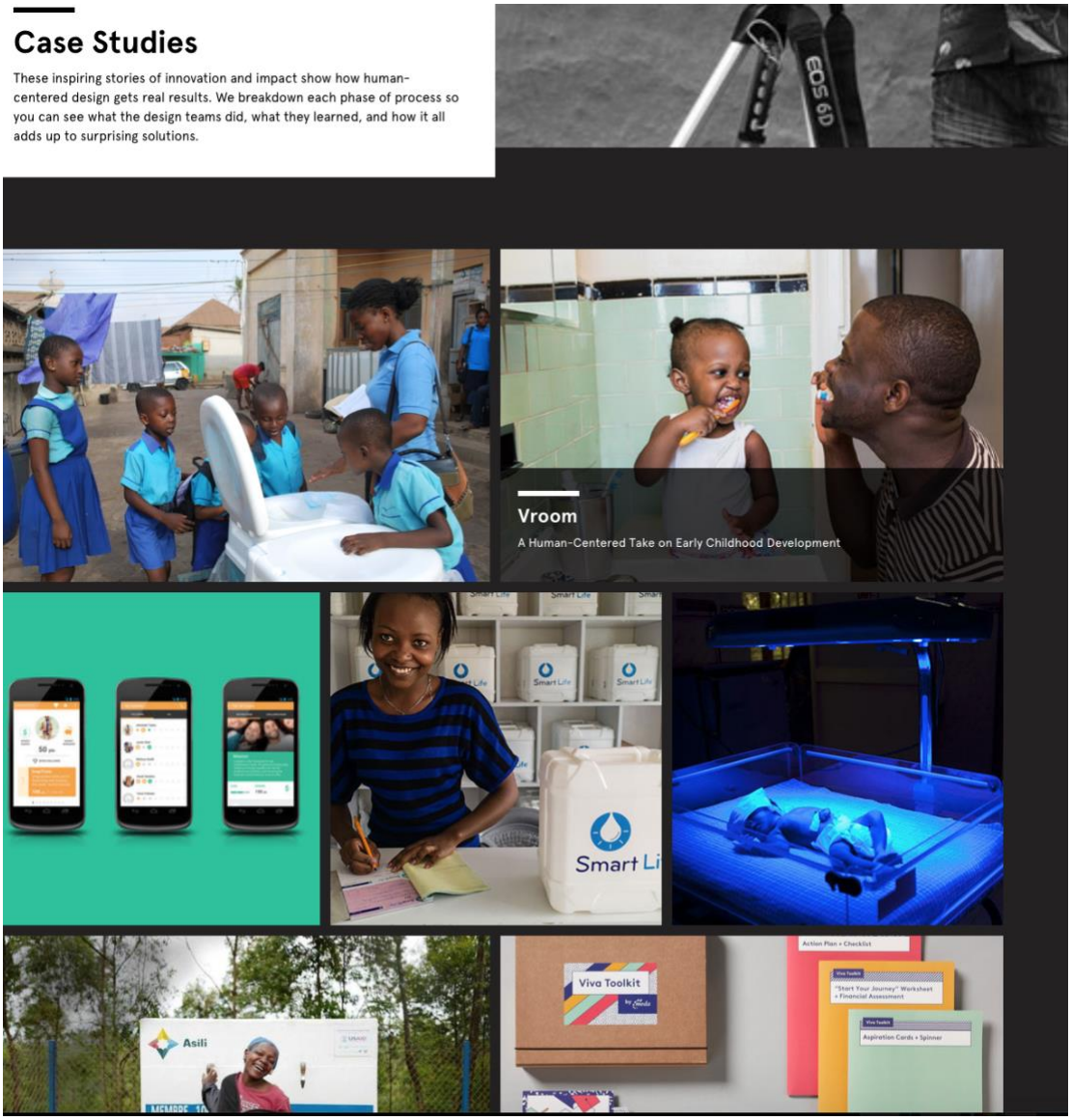


Figure 29: Examples of cases where UCD methods and processes have been used[8]

[8] https://www.designkit.org/case-studies

Summary design toolkit research

Design Requirement	Met by existing toolkits?	Design features included
DR1: Need to support multiple practices	Yes	Technique descriptions (when, why where)
DR2: Needs to be perceived relevant for practitioners	To some extent	Present challenges Examples of use Examples of outcomes
DR3: Needs to change alongside practices	No	----
DR4: Needs to meet the specific challenges faces by practitioners	To some extent	Examples of use Present challenges (Figure 27) Details from methods (Figure 28)

Table 7: Design requirements and existing toolkits

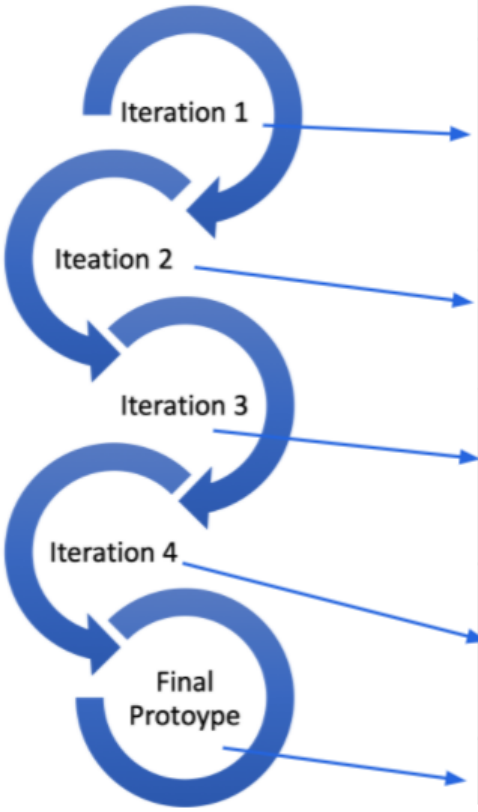
Through the research of online toolkits, I gained insight into possible solutions and design features that could be explored further. Examples of this were expected outcomes, "quick" stats like time/materials needed and challenges specifically for phases. None of the presented toolkits have "all" the design requirements met as seen in Table 7, something I am aiming for in my design

6.2 Wireframing – the development of the prototype

Wireframing was the main type of design work done to conceptualize the prototype presented in this thesis. The design work done in four main iterations, an initial exploratory iteration, an iteration focusing on system architecture, an iteration for exploration of interaction and flow and the final iteration which had as an aim to develop a prototype for evaluation, which was influenced

of the other iterations as well as theory and empirical insight. As mentioned, I will structure the sections of each iteration, is structured accordingly: insight from previous empirical activity (evaluations/diagnostic work), the prototype at said stage and the results from evaluation. Each iteration was also lead by an overarching question which was the focus of the iteration(Figure 30)

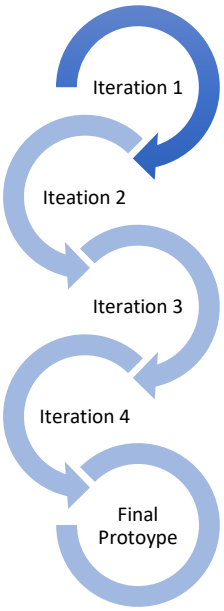
Each of the iterations in my design process are presented by one row in the figure below (Figure 30)The empirical insight gained from evaluations and design has combined with the diagnostic findings been crucial in the establishment of design principles as they are so closely tied to the artefact itself and the practices of the DHIS2 practitioners.



Focus of iteration	Central iterative question	Key insight
Initial exploration of prototype and form	What content is needed and how should it be presented?	There are many different methods utilized by practitioners. Existing toolkit do not meet all requirements.
System architecture and logic	How can the prototype support scaling and change?	Practices changes, there needs to be flexibility present.
Aesthetic properties of content and form as well as form	How can the structure established in iteration 2 be presented and what is required from it?	Not all methods seem relevant - when, how, what, why and where needs to be stated
Add design features and get it ready for evaluation	Can this concept support and promote UCD? Do the practitioners see the utility of the prototype?	There is an array of challenges and solutions that are similar among DHIS2 practitioners
Derive design principles and evaluate utility		

Figure 30: Iterative design process

6.2.1 Iteration 1: Initial exploratory iteration



Insight leading into iteration:

The design of the first prototype was influenced by the diagnostic field work in Malawi and Mozambique (4.3.2). Specifically, the presence of multiple methods and the insight that these needs to be supported. The insight that there was a need for some type of resource was also one of the key driving forces for this iteration.

Prototype*:*

The design work was focused on what was needed when presenting methods, guided by questions like what are the methods, what are the aims and why should you do the presented methods? Some examples of first prototype is presented below (Figure 31). As seen in Figure 31, some of the prototyping was more conceptual in its form, allowing me to explore aspects including what should be included when and where in the prototype.

Wireframing

Practicalities:
Time frame: 0.5 - 4 hrs
Materials needed: Software / paper & pen
Participants: 1 - 9 people

WHAT:
A wireframe is a mock-up of a system/app/service. It is a two-dimensional illustration with no real functionality.

When:
Throughout the development process.
Both early and late.

WHY:
It is done to illustrate and communicate the properties and functionality of a system.They are useful to get input from the user as well as communicate ideas within the team

Combinable with:
Scenario, personas, guerrilla usability testing, usability testing.

STEP 1:
Build preliminary blueprints that show structure, placement and the hierarchy for your product
STEP 2:
List what UI/UX patterns you want to use
STEP 3:
Review wireframes with specific personas and scenarios
STEP 4:
Discuss wireframes with team to get feedback on structure.

Investigation

WHAT:
Investigation is concerned with gaining insight into users, the context they use technology or do activities in, as well as what they know, don't know and skills they have. This activity relies heavily on a need to talk to users and go into the "field".

AIM:
By gaining insight into the users, we as developers and designers are able to better design systems/apps/services that are relevant and usable for the users themselves.

WHEN:
This is done throughout the process. Initially is is important to get to know your users, their activities and their context. Later in the process you might want to test your system or there is a need for more insight and this activity can be done again.

Techniques:
Observation, interviews, surveys

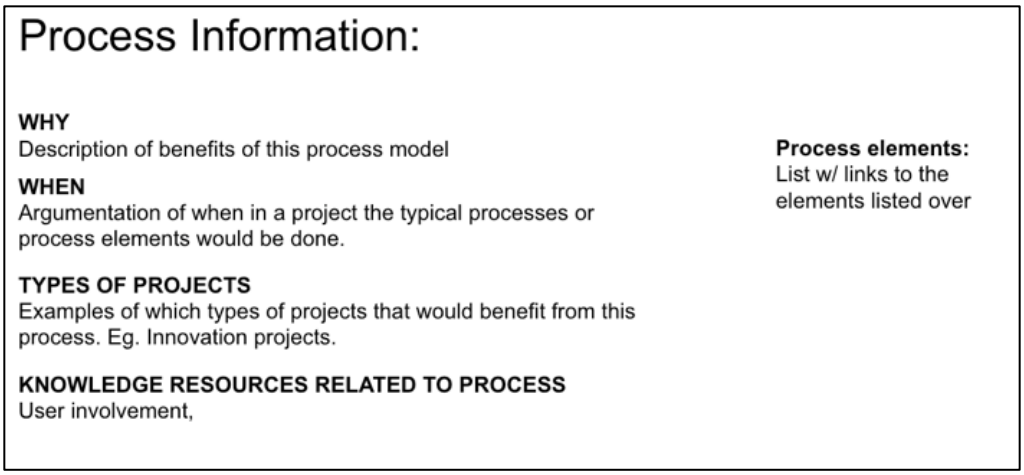


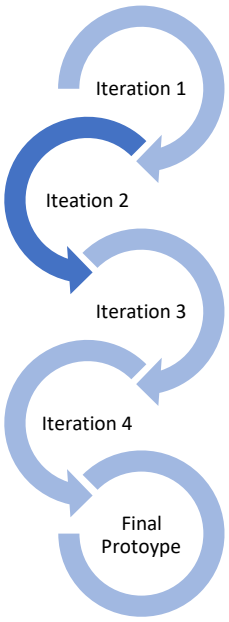
Figure 31: Early prototypes of method toolkit

Results from evaluation:

In the figures above, examples from the first prototype is presented. Through co-analysis with fellow master students of the Design Lab, I gained feedback from the most successful ways of presenting information. This is presented as a "design feature" below:

Design feature
Need to establish what the aim of a method or technique is, e.g. to establish that the aim of "investigation" is to understand users and their context.
What practicalities are needed to conduct said method/technique.
How to conduct said method/technique (e.g. steps in Figure 8)
When in a process a method/technique is most ideal or provides sufficient insight.

6.2.3 Iteration 2: System architecture and logical structure.



Insight:

Needed to establish a logical structure so that it is easy to add, remove and change methods and techniques as we learn more about practices and projects. This became apparent the more the design was explored in iteration 1 as the goal was for this prototype to be an ongoing, collaborative project. A standardized template of what elements should include as well as the system itself, could make the takeover by other designers or developers easy. This was further underlined by the empirical insight gained during the diagnostic activities, as projects change calling for new types of methods, processes and techniques.

Prototyping:

During this phase, the internal logic of a method system was established based on the initial design from the previous phase. The relationship between elements was prototyped and sketched up. The modular hierarchy of the system was prototyped through sketching as seen in Figure 32.

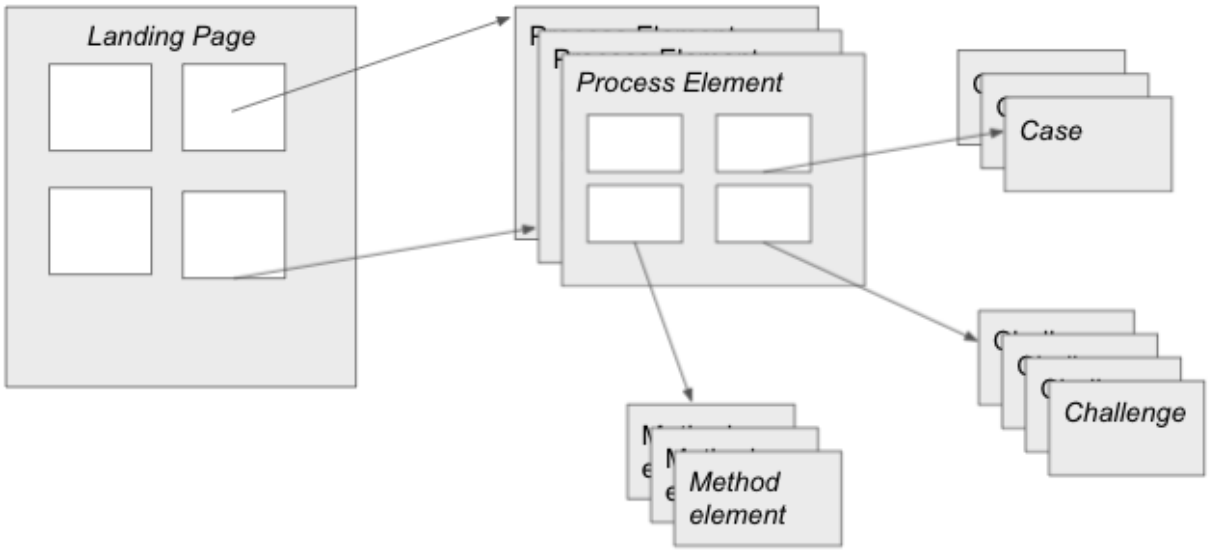


Figure 32: Hierarchy of system

As seen in the figure above (Figure 32) there are five major elements in the modular system, the landing page, process element, case, technique and challenge. The landing page is the "hub", which links to process elements, these being larger phases like "insight" or "prototyping". Each element should contain appropriate techniques like interviews or wireframing (method element), and the challenges are challenges associated with said process element.

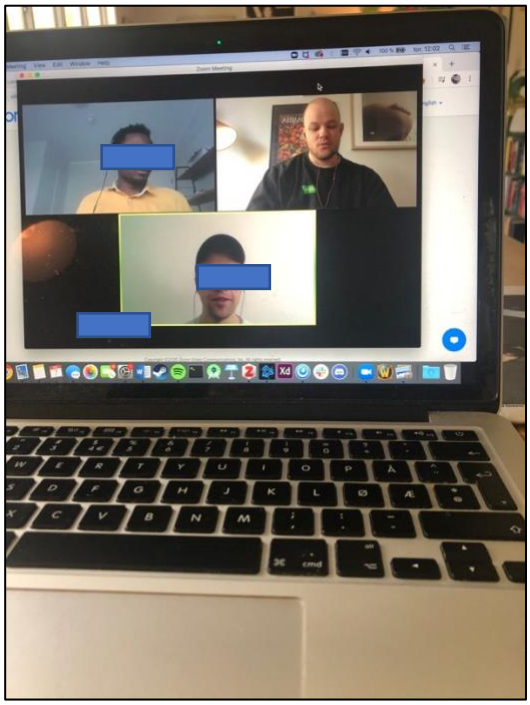
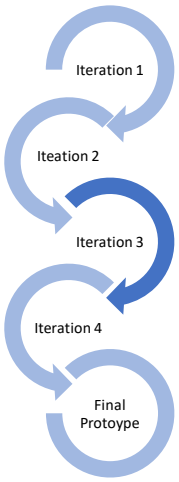


Figure 33: Virtual Evaluation with developer

Evaluation

Feedback on the structure of the system was crucial as the expert pointed out the feasibility for maintenance and flexibility of the prototype. The evaluator noted that the flexibility and modularity is important for the growth of the system and allowed for change in one part of the system not to affect everything in the others (high cohesion and low coupling).

6.2.3 Iteration 3: Design of interaction and flow



Insight:

Gaining positive feedback on the structure established in iteration 2, the aim was to explore how this structure could be presented in the prototype. A key insight from the virtual interviews that was apparent in this iteration was that a significant portion of the UCD methods supported are not seen as very relevant to the practitioners, and there is uncertainty of when, why and how it is appropriate to use the different methods, which a toolkit like this need to support. Another key insight was from the research of toolkits, inspired by their presentation of methods and techniques as well as how some design requirements found from the diagnostic work was not well covered by the existing toolkits.

Prototype:

The prototype was influenced by the structure established in iteration 2, where I wanted to explore how this type of structure could be presented in an interactive prototype. The flow specifically was

explored in the instantiation of the model developed in iteration 2 (Figure 32), through simple buttons and hierarchies within the system.

The different aspects of the prototype that was established, and aesthetic exploration of the landing page presented in Figure 34 was started.

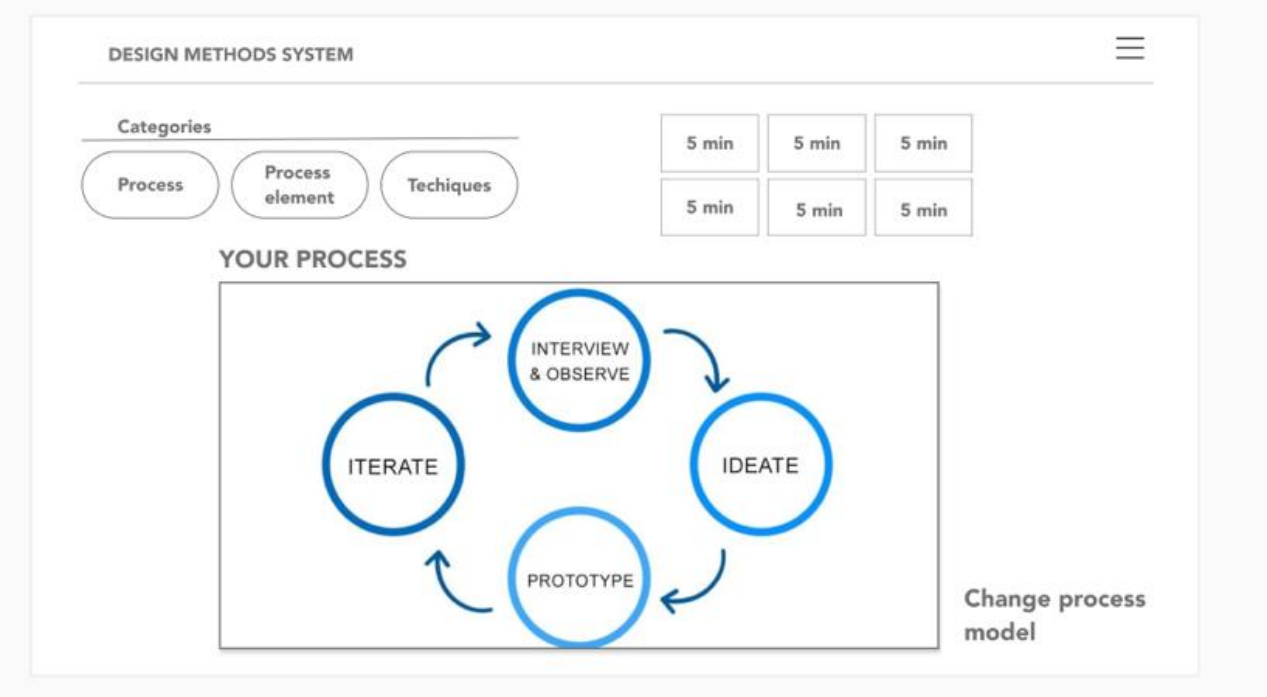


Figure 34: First iteration of "landing site", method toolkit

Under this page I developed what would be called a *process element*, which would be major activities in a “typical” user centered process as well as inspired by empirical insight gained during previous and ongoing diagnostic work. Examples of a process element would be investigation, problematizing and prototyping, following the phases presented in the figure below (Figure 35).

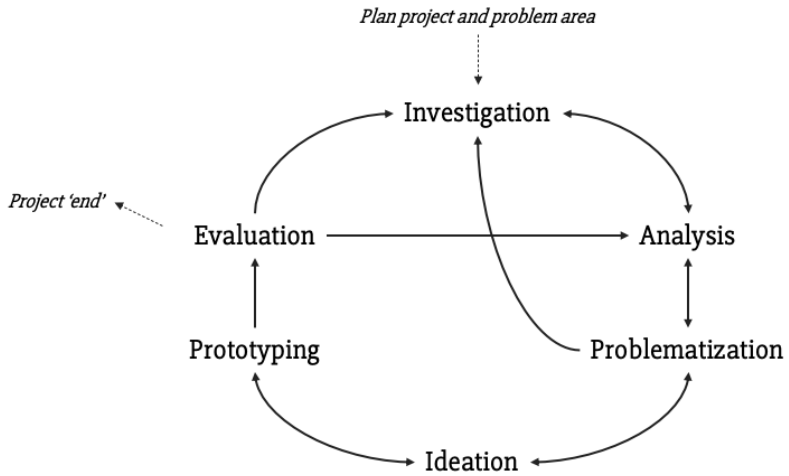


Figure 35: Configuration process that emerged from empirical insight

The techniques, the smallest parts of the system were explored further in this iteration. Many of these techniques were investigated in the previous power point prototype (Iteration 1), however in this iteration, more examples were added from empirical work, as well as cases from DHIS2 implementation where the techniques had been used were added, inspired by the toolkit research (Figure 29).

Instantiation of the design requirement gathered from toolkit research and diagnostic work previously presented was explored further, as seen in the figures below (Figure 36 & 37). Designing the techniques, I tried to meet some of the challenges found during the diagnostic work, like how to promote relevance through presenting how and when techniques are appropriate as well as what other techniques that "fits" well with the current element. Through this, the idea was that the users of the toolkit can easier plan to use or conduct the techniques and methods.

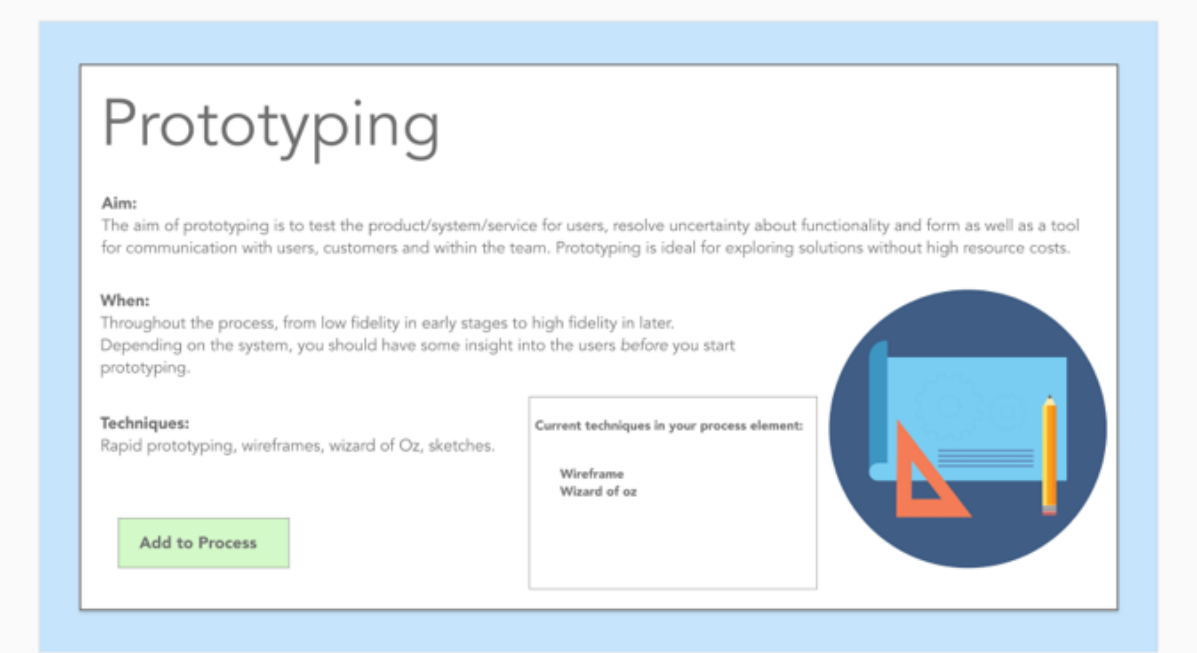


Figure 36: Second iteration of content – "process element"

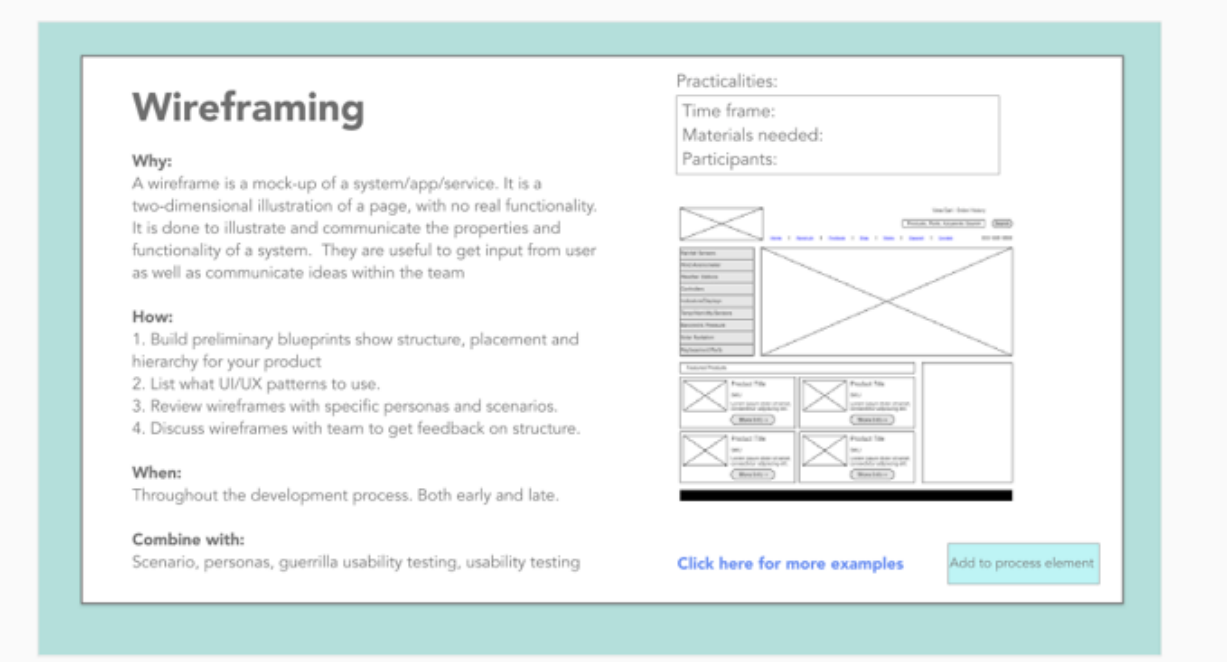


Figure 37: Second iteration of content, "technique"

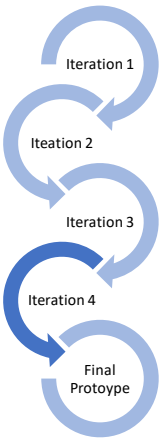
Evaluation Results:

In this prototype, the idea that one could "build your own" process to fit specifically for a project was explored seen in Figure 37. This being one of the solutions to overcome the perceived lack of relevance for process models and methods in DHIS2 projects. However due to change in the participants of the designers, technical and time constrains, this idea was not explored further at this point.

Below is the previous list of design features, but with added design features that followed into the final phase of design work, marked in green:

Design feature
Aim of method or "phase" (e.g. investigation)
What practicalities are needed to conduct said method/technique
How to conduct said method/technique (e.g. steps in figure 2)
When in a process a method/technique is most ideal or provides sufficient insight.
Examples of projects when methods are used
Hierarchy to help the logical structure
Settling more on how elements should be presented

6.2.3 Iteration 4: Design of final prototype and evaluation



The final design phase of my thesis was started in the fall of 2020 with the aim to develop a prototype that could be evaluated by practitioners.

Insight

Design work and diagnostic work happened very much in tandem during this design phase, where one diagnostic activity (e.g. interview) would directly inform the design method toolkit leading to a design change. During the co-analysis work mentioned over, I would both analyze and design at the same time, allowing for quick changes in the design.

In the final design phase, many of the design decisions that would be important to the system and subsequent design principles emerged. These specifically being challenges that emerged during the diagnostic work as well as solutions to these. As mentioned above (thematic analysis 5.2.2), many of the challenges that emerged in the diagnostic work were similar from project to project, leading to me wanting to explore how this would be represented in the design of the artefact, an idea taking inspiration from the previous research into design toolkits.

Prototype

This is seen in the example below (Figure 38) where the challenges around analysis was explored and presented in the prototype.

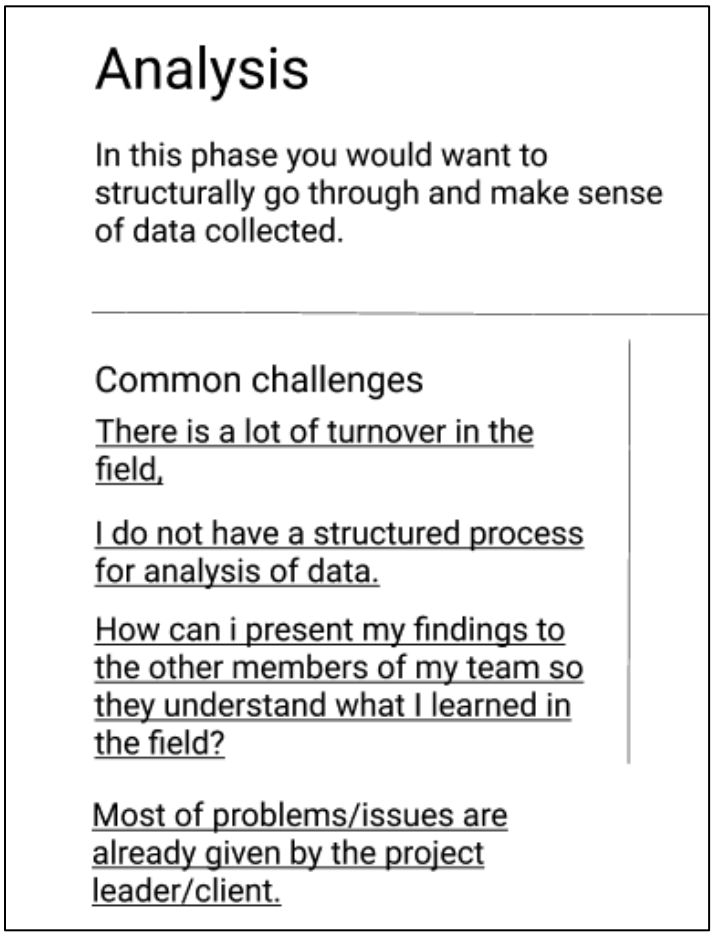


Figure 38: Example of analysis challenges

When asked why or why not UCD methods both were and weren't used, the challenges identified during the diagnostic work were used as examples. Therefore, I wanted to explore if these could be explored as a way to increase the perceived relevance of the resource itself. If challenges to applying a method stem from lack of knowledge or expertise with these, I wanted to explore if this could be met and supported with a resource like this.

Using my own design knowledge and collected empirical insight, I explored how solutions to the challenges presented could be met as well during this design phase (Figure 39).

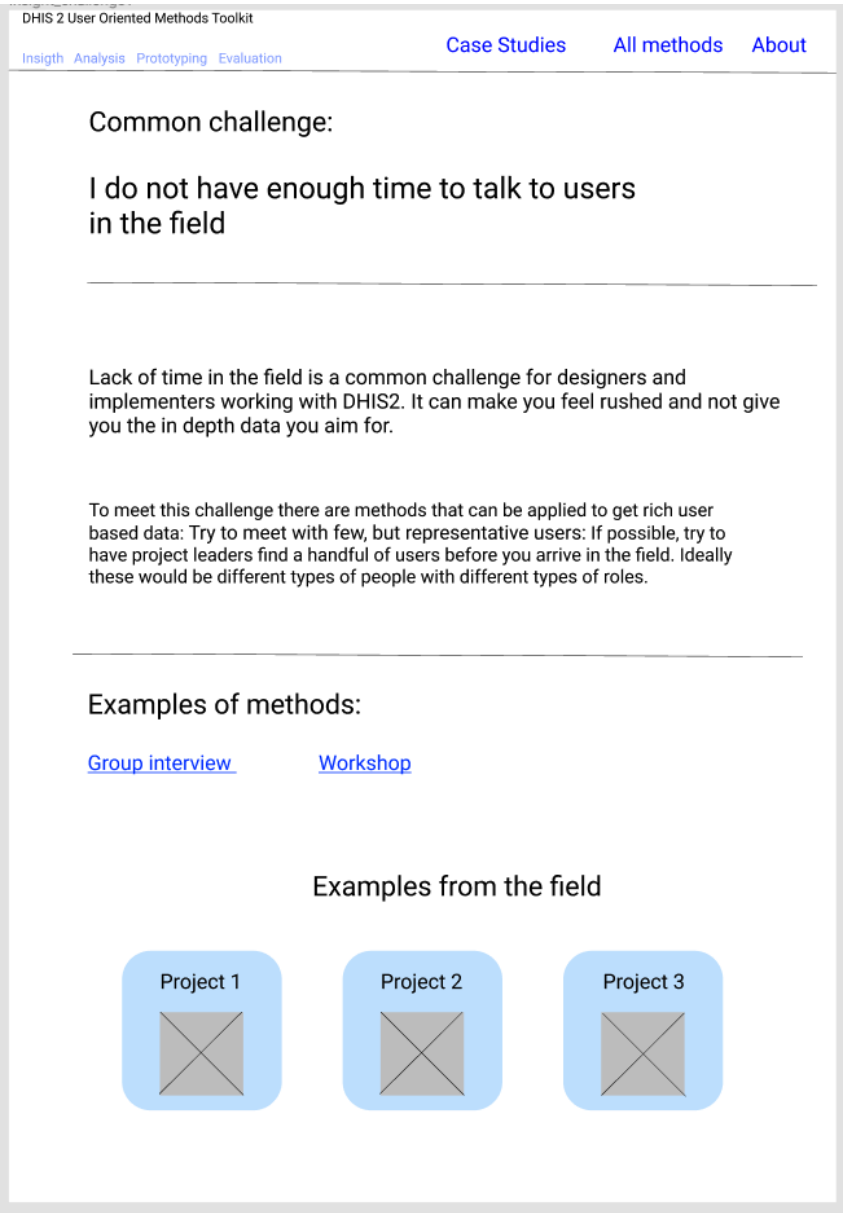


Figure 39: Example of "solution"

During the final iteration, interaction was added to the prototype, allowing be to send it to practitioners allowing them to interact with the prototype on their own terms (Figure 48)This was represented in the landing page (Figure 40) which was further developed in this prototype, allowing for quick recognition of processes and movement on the site by pressing in process element.

Evaluation

During this iteration, I had two separate evaluations, making this an iteration with two "minor" iterations within, one concerned with form and concept and one more attuned to the content. See **4.6 Evaluation** for more details on how this was conducted.

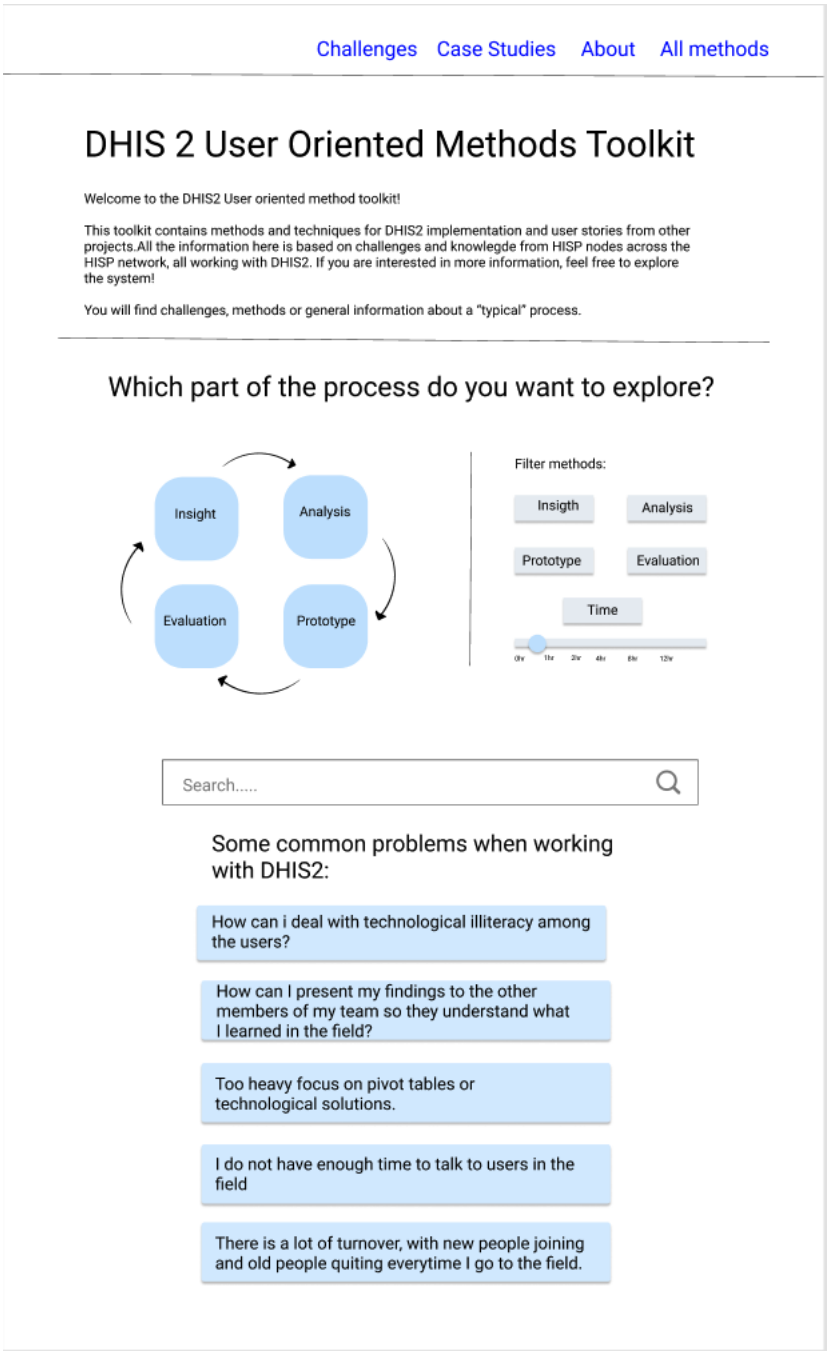


Figure 40: Screenshot of landing site, method toolkit

The landing page was one of the pages that got the most feedback during evaluation and saw the most change. The participant in the first evaluation did not recognize the process, as the participant noted, *"it would be better if you added some smaller activities between the phases".* The participant also wanted more information as the first iteration of the landing page did not convey the purpose of the toolkit in a sufficient manner. The presented page is the final version, but between the first and second evaluation multiple features were added:

- Search bar.
- Description of what the toolkit is.
- Some challenges that were perceived as especially important.
- Filtering.

The utility and relevance of the toolkit gained positive feedback during the final evaluations, for example when asked whether or not the participant could see themselves or a colleague using the toolkit they said; *"yes, I would see myself using a tool like this*", further elaborating *"if you have a guide like this, it would be nice to make an implementation process".*

When asked whether they would find it relevant to try conducting some of the techniques presented without previous experience in them in an implementation process, they replied; *"oh, it explains step by step what you need to do, from one activity to another".* The participant also underlined the familiarity of the methods and especially challenges presented, and they could recall situations where they had meet these exact challenges or challenges like the ones presented.

The design features that were added to the prototype during the final design phase is marked in orange in the list below.

Design feature
Aim of method or "phase" (e.g. investigation)
What practicalities are needed to conduct said method/technique
How do conduct said method/technique (e.g. steps in figure 2)
When in a process a method/technique is most ideal or provides sufficient insight.
Examples of projects when methods are used

Hierarchy to help the logical structure
Settling more on how elements should be presented
Added common challenges as a major design feature
Added solutions to the challenges to increase relevance
Adding header bar to ease navigation
Interaction and flow added to help virtual evaluation
Explore options for filtering
Search bar on landing page

6. 3 Summary of design work and results from evaluation of final prototype

The design process presented in this chapter along with the diagnostic findings and feedback from the evaluation has been the basis of the design principles presented in Chapter 8. The next chapter presents the final version of the prototype at the point where the empirical work ended for this thesis, but the design work on the toolkit are currently being continued by other master students in the DHIS2 Design Lab.

CHAPTER 7: Artefact description

The following chapter will present the artefact (DHIS2 design method toolkit) and its features.

7.1 Landing page

The landing page (Figure 41) of the toolkit is the first "meeting" with the toolkit. The toolkit is focused on the four "major" phases in a typical user centered process as presented in 3.1.2. These four phases, through some semantic differences are represented in all the empirical work and insight I gathered throughout my process, thus can be represented in some way in most projects investigated. The landing page has filtering on these said phases as well as the amount of time specific activities need.

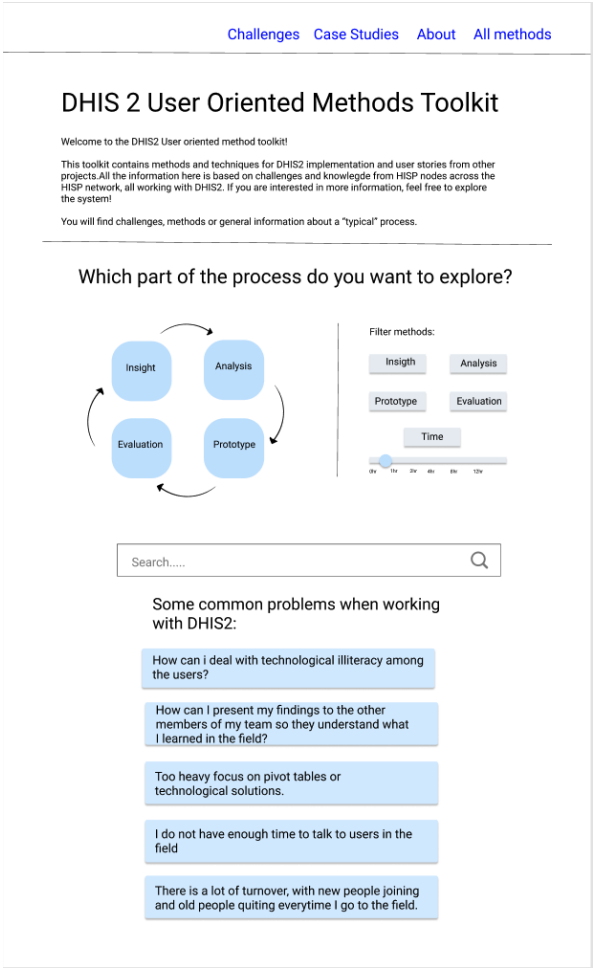


Figure 41: DHIS2 method toolkit landing page

Some of the most common challenges experienced by the implementers is also present, for future work can these be presented according to popularity. All the phases and challenges are also clickable, leading to subpages with more details.

7.2 Process elements

Every process element (the phases) has a page connected to it, working as a “hub” for challenges, methods and examples. As seen below (Figure 42), this page is designed to provide the user with an overview of the phase, what it is and what it includes.

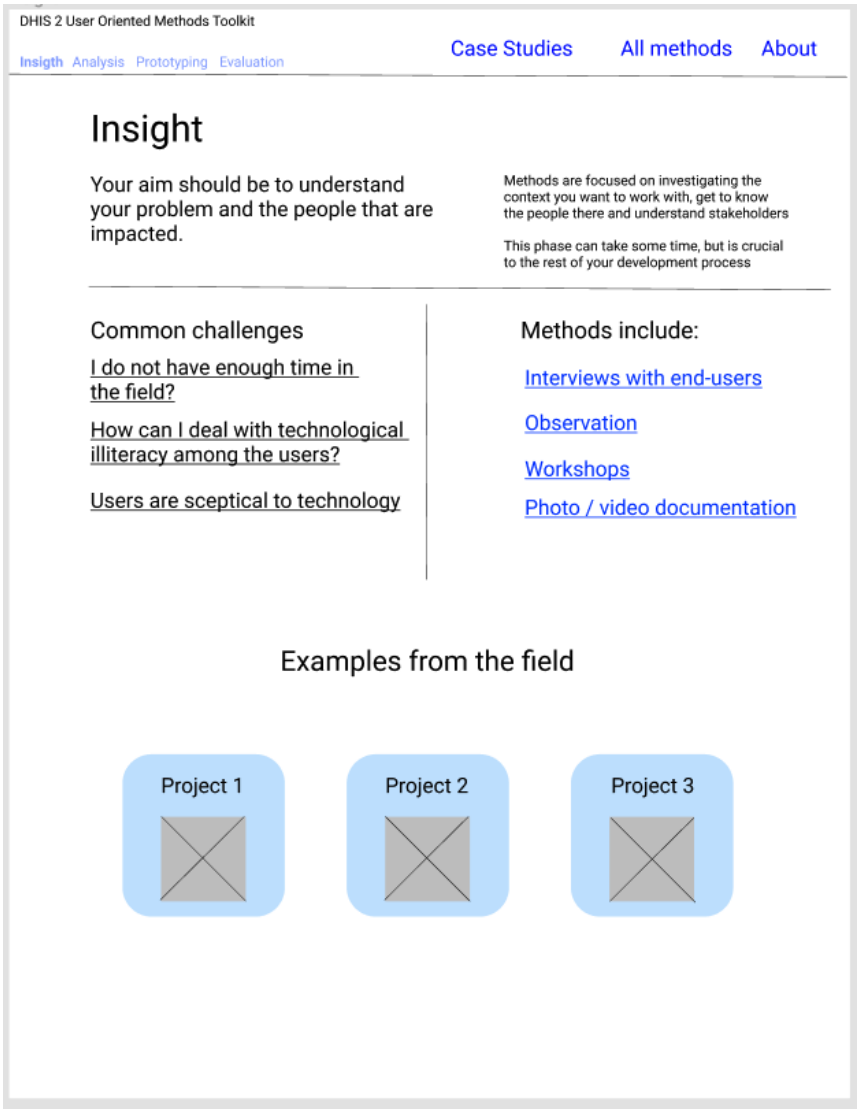


Figure 42: Example of "process element"

In each element there are common challenges, emerging mostly from my empirical work as presented in chapter 5. These are interactive, thus clicking the "challenges" will link you to sub-pages in the system centered around a solution to this specific challenge.

Methods specific to the process element is presented through links to other pages (see below). It is important to note that several methods can be presented in both several process elements as well as several solutions.

As the DHIS2 Design Lab gains more experience and insight into projects across the HISP network, more examples can be included for each of the process elements.

7.3 Method element

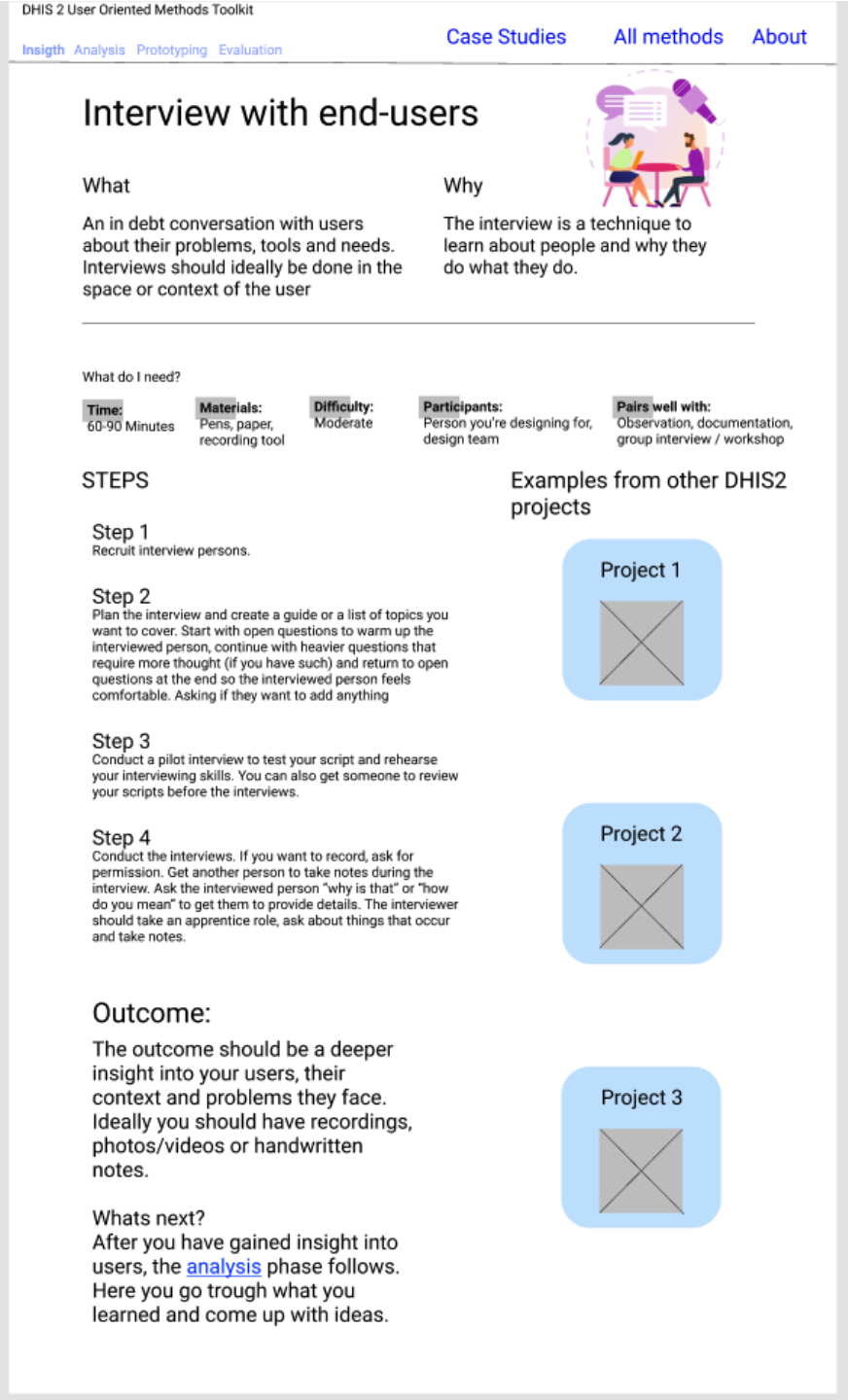


Figure 43: Method element

Each of the methods in the toolbox have specific pages as well, referred to as method elements (Figure 43). These consist of the steps needed to perform said activity, explanations of why and what a method is, stats of what is needed to perform it, examples of DHIS2 projects where the method has been used and expected outcome and "next step" in a typical process.

By providing links from method to method, and through examples, the user should be able to follow a process and thus get guidance throughout the planning of projects.

Each method follows the template of what the method is in general as well as why one should do it. This is so that practitioners can research and decide to use the method that works the best for them.

Some quick information is also present in the method, what are named as “stats” in the system (Figure 44). These are what is needed, from materials and time to what other methods the present one pairs well with.

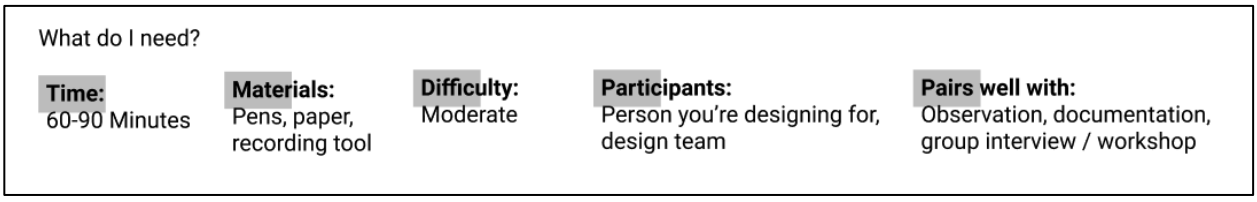


Figure 44: Example of quick stats for each method

7.4 Challenges and solutions

The concept of challenges and solution to these challenges is one of the pillars of the method toolkit. Based on the challenges identified in the diagnostic work the presented challenges in the toolkit reflect specific challenges DHIS2 practitioners meet in their work with and for users (Figure 45). This is a key part of the toolkit as these challenges are central to what sets the toolkit apart from other method toolkits on the market as they specifically are concerned with the domain, not the methods. Thus, ideally provides utility specific for DHIS2 practitioners.

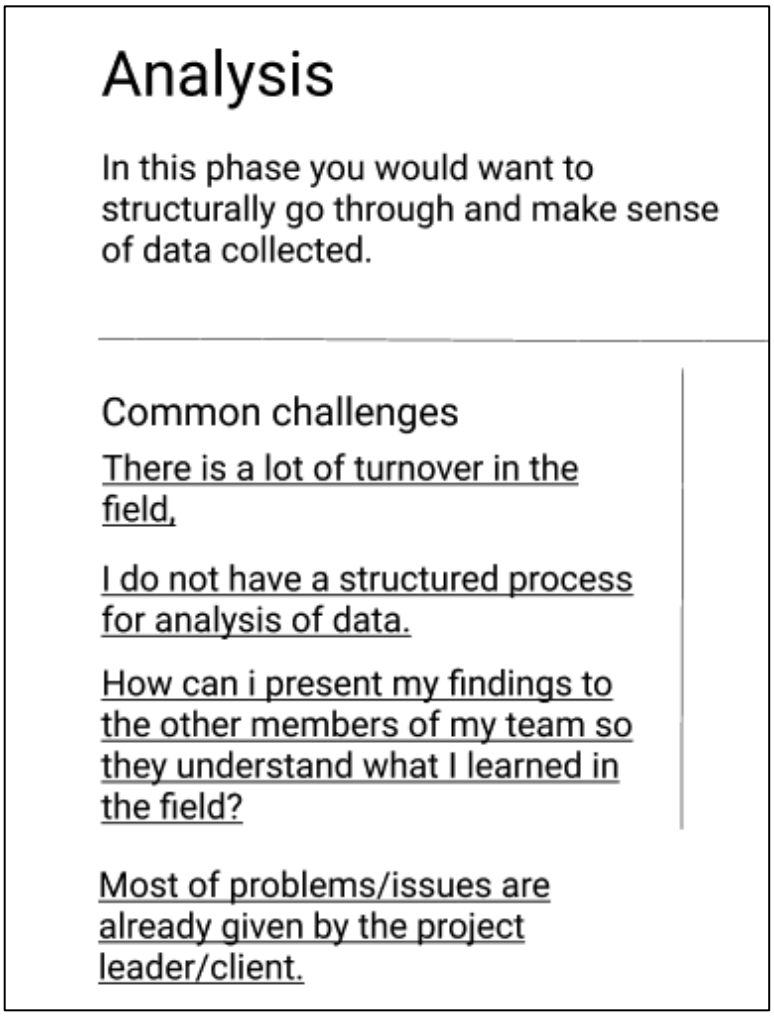


Figure 45: Example of common challenges

As the example demonstrates each process element has their own unique challenges. This is presented and designed so that the user easily can get and overview and hopefully find the same challenge they have themselves. A major finding from empirical insight and design work was that throughout the DHIS2 ecosystem similar challenges with user centered methods and design are present. For example, not enough time in the field or a too heavy focus on pivot tables.

By clicking on the challenges, the user is presented with one solution to the challenge at hand. These solutions are a combination of empirical solutions to challenges found through observations and interviews as well as my competence in user centered methods and design.

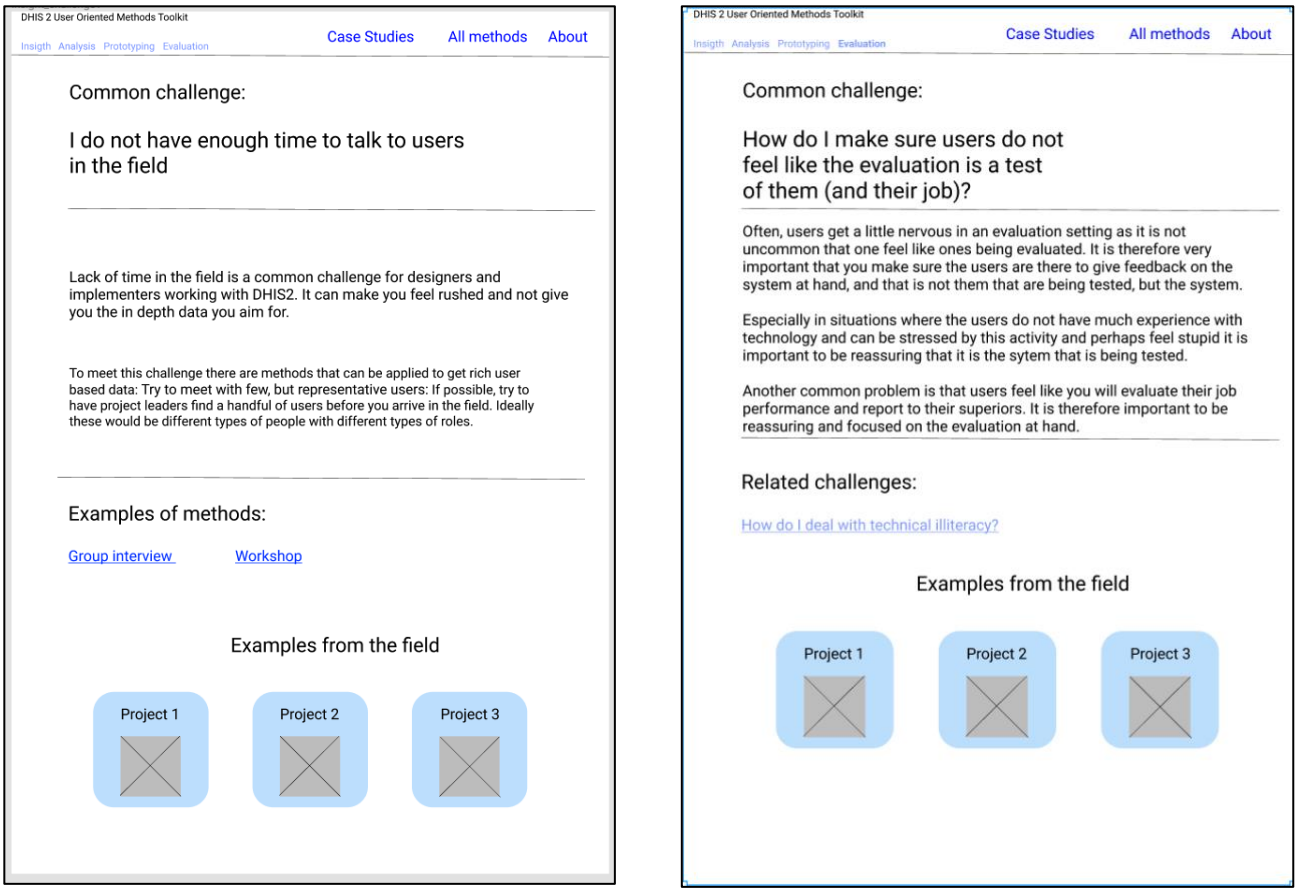


Figure 46: Example of solution to common challenges

7.5 Navigation

A key aspect for the navigation of the toolkit is the possibility to move around in the toolkit itself. This is to explore methods and phases for different types of projects and different types of needs. To help this be realized, I added a header bar that is the same throughout all the pages of the toolkit (Figure 47).

Figure 47: Navigation bar of the toolkit

This header enables quick and efficient navigation through the toolkit, with the process elements (typical phases) to the left and the methods and case studies to the right. The header is crucial for the flow of the prototype as presented in Figure 48.

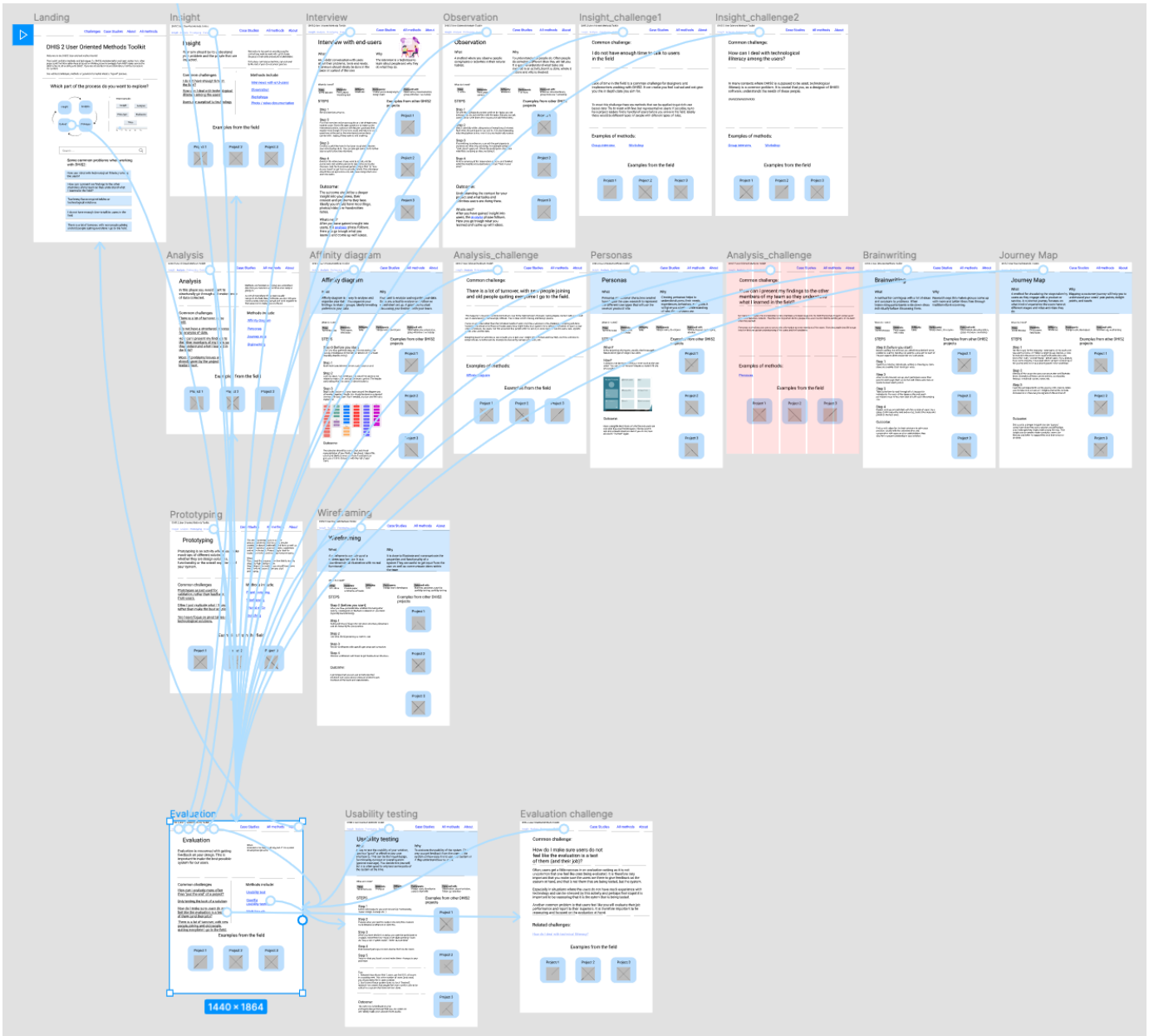


Figure 48: Flow of prototype

7.6 Summary

This chapter presented the final artefact of my thesis, the DHIS2 method toolkit, and its key features. The toolkit is the culmination of the diagnostic work and the design and evaluation presented in Chapter 5 and Chapter 6. The toolkit is an instantiation of the design principles presented in the next chapter (Chapter 8), and the design have been actively used to derive these.

Chapter 8: Discussion

In the following chapter I will present my design principles. They are mainly derived from empirical insight and I will discuss the principles with literature visited in chapter 3, before I discuss the contributions and implications of my thesis. To revisit the introduction, the research question that have guided this thesis is:

What are design principles for a resource to support and promote user centered design during implementation and configuration of generic software?

This question tries to address both the case specific issue of lack of support in conducting user centered methods among the various HISP groups, and contribute to the IS literature concerning how support and promotion of user centered methods in generic software ecosystems can be done. In my case I argue this can be done with the creation of resources, like the toolkit developed in this thesis, that are specifically grounded in the work and domains of intended practitioners, making these feel familiar and relevant to their practices. The presented design principles in this chapter are designed and presented in a way that vendors that aims to promote and support UCD methods in their respective ecosystems can have some guiding principles to what such a resource could be.

I will articulate and discuss the design principles in the following sections. The principles are summarized in Table 8**.**

Design Principle	**Specification of principles**
DP1: the toolkit should be mutable	The toolkit should be designed in a way that the implementers easily can add, remove or change content due to the changing nature of their practices.
DP2: the toolkit should present a diversity of methods, processes and techniques	There should be more than one method present for same types of activities. With methods and techniques integrated in the resource from both

	empirical accounts of practice and theory from UCD.
DP3: the toolkit should present readily recognizable examples of common challenges	Challenges specific for the domain and the users it is developed for should be present in the toolkit. Ideally these should be met with solutions (DP4).
DP4: the toolkit should present examples of solutions to challenges that are readily recognizable by designers	The toolkit should have solutions to the challenges presented in DP3. These can be from other practitioners in the same ecosystem/domain or based upon the competence of the resource designer themselves.
DP5: the toolkit should have a level of abstraction that readily meets users' mental models	The toolkit needs to be familiar to the practitioners it is designed for. It should be based on their language, their processes and their methods. However, it is important that if it should include multiple groups/domains/organizations, that a "common" yet recognizable mental model should be established.

Table 8: Design principles and specifications

The literature reviewed in this thesis is mainly concerned with the promotion and support of user centered methods. The lack of perceived relevance of the methods, their lack of "fit" in a context and their resource draining cost were challenges found in the literature (Ardito et al., 2014; Bang et al., 2017; Otkjær et al., 2008). These challenges concur with the empirical findings of this thesis as presented in chapter 5. Further, the literature argues for the lack of utilization of suitable UCD methods that fits into software engineering practices is a major cause for usability issues and failure of large projects (Gordon Baxter & Sommerville, 2011; Dittrich, 2014).

The following Figure (Figure 49) summarizes the design principles which will structure the following chapter.

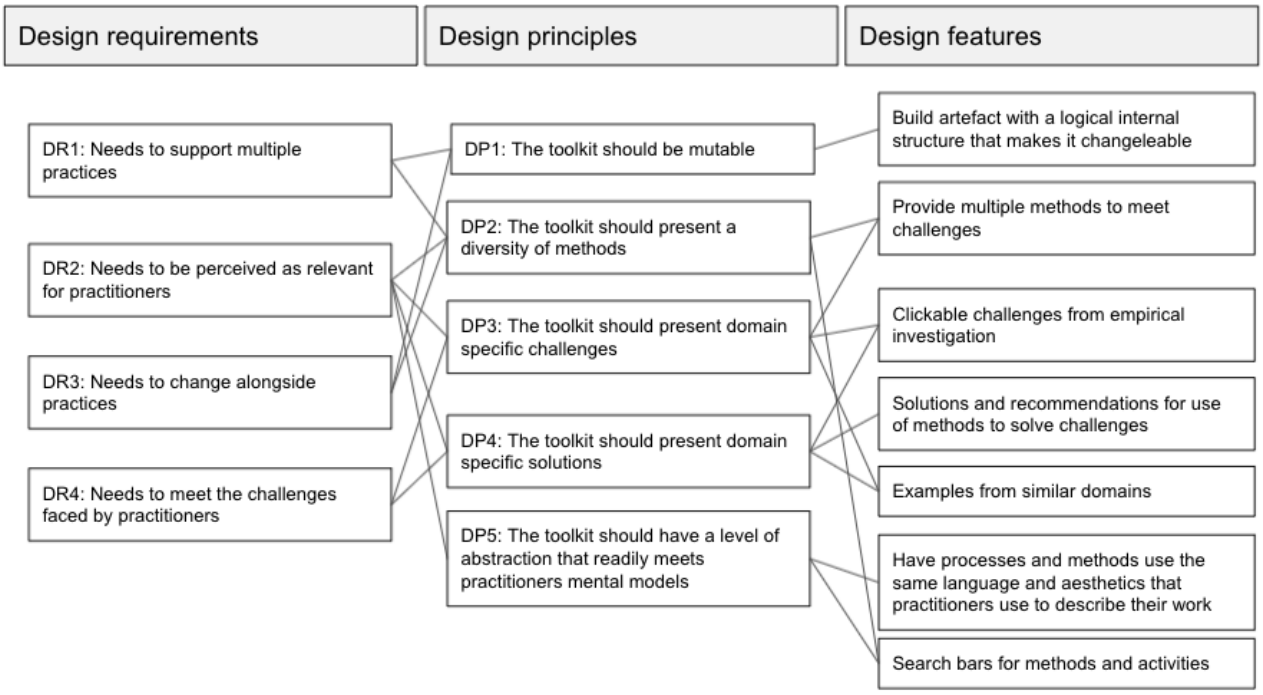


Figure 49: Connection, design requirement, design principles and design features

8.2 Design Principle presentation

This section will discuss the design principles derived from the DHIS2 design method toolkit and discuss these with the literature presented in chapter 3. It is important to note that the principles are artefact focused as opposed to process focused. They are concerned with aspects of the artefact itself rather than the process of designing it.

Design principle 1: the toolkit should be mutable

Methods, processes and practices changes over time, and are not static instances. This implies that the resource needs to be designed in a way so that the users of the system easily can add, remove or change content due to the changing nature of practices from project to project. As well as adding and removing content if they find some methods more suited to problems or inefficient in specific situations.

As seen in my empirical work, the nature of software projects related to DHIS2 significantly differ. From fully new innovation projects such as ePompar (in Mozambique), a unique software developed for community saving, to projects mainly centered on digitization of hospital paper

records. These projects need to be supported in different ways, as new methods and processes emerge, underlining the need for a mutable artefact.

This can be achieved by designing a stable and robust internal structure, through a system design based on set hierarchies and internal roles (Figure 32). By doing so, it can be changed and expanded on over time, as new empirical insights and practices change. Adding and removing methods, techniques and changing processes. Following Baxter and Summerville (2011), they argue that methods for user centeredness have not changed and kept the pace along with the markets they are used in, thus designing a resource that is changeable along with practices might help meet this problem.

Design principle 2: the toolkit should present a diversity of methods, processes and techniques

As seen in my empirical findings, the DHIS2 practitioners utilize many different methods for the involvement and collaboration with users. There is a diversity of unique projects, all with different approaches, techniques and methods. This diversity should be present in a resource, to add to the familiarity and perceived relevance of the resource.

Often, the techniques promoted in methods like user centered design, does not "fit" well in the context DHIS2 are being implemented in (Teka et al.,2018). This is reflected in how the investigated software development practices differ from the smaller bespoke projects most of the examples where user centered design is promoted are. The cultural context of DHIS2 implementation often differs from that of UCD examples, as they often are Eurocentric, something DHIS2 implementation is not. Like the DP1, the argument of Baxter and Summerville (2011) is added to in this principle to, and acknowledges that methods change across time and space often informally, which should be supported by the presence of multiple methods in the toolkit itself.

This is mirrored as well by the challenges pointed out by Teka et al. (2018), where they noted that the often promoted user centered methods do not fit in the cultural context they are promoted in. This was apparent in my own diagnostic work, where many of the practitioners involved in the study noted that many of the methods promoted through academia do not fit in their contexts. This might be due to different hierarchical structures or accessibility to users. Through a presentation of an array of different methods, I would argue the chances for finding a method that "fits" might

be more accessible, leading to a possibility for more user centered work. This logic is very much in line with that of discount usability methods, as presented by Nielsen (1995), where "some usability is better than no usability". During artefact evaluations, one participant also noted, when presented with the solutions, the challenges and the methods in the toolkit; *"different possibilities for solutions might be nice, not everything fits all".* This helps illuminate the relevance and importance of having a diversity of methods present.

Design principle 3: the toolkit should present readily recognizable examples of common challenges

Central to this principle is the presence of challenges common to practitioners in the domain the resource is intended and design for. Ideally these common challenges should come from the practitioners themselves, to make the challenges readily recognizable, increasing the perceived relevance and utility of the resource. These challenges should be clickable or link you to information to overcome the challenge itself (see DP4). The picture below (Figure 50) is from the toolkit developed in this thesis.

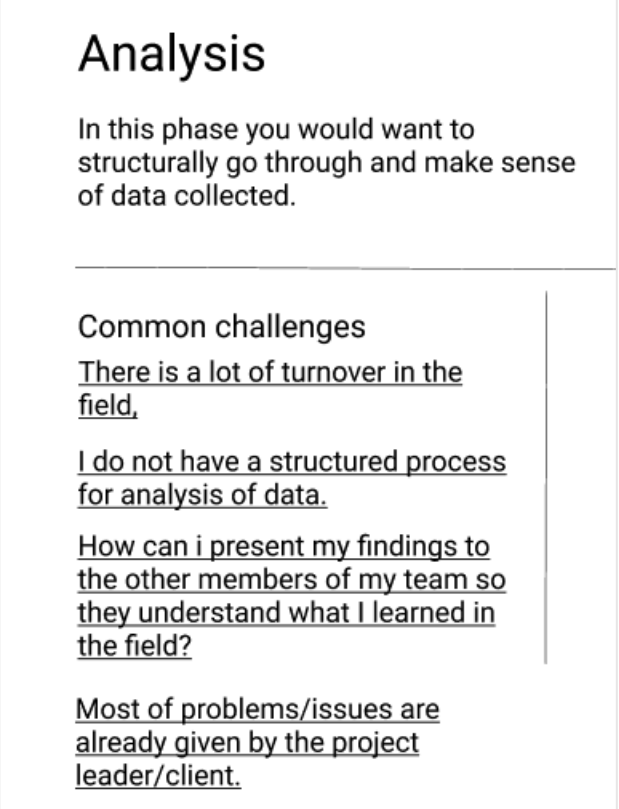


Figure 50: Example of DP3, common challenges

Howarth et al (2009), presented scaffolding as a successful premise in their development of a resource to help usability practitioners in their work. This was represented in the presence of challenges and solutions to specific challenges in a domain. This finding was very much replicated in the findings of my study as well. During evaluations, the challenges concerning user centered methods from across multiple DHIS2 practitioners were reviewed as one of the most accessible design features for support of user centered methods. As one practitioner noted during evaluation: *"Yes, if I come here and I see whatever it is I am struggling with, it is much easier to see the proposal for a solution."*

An argument posed by Teka et al. (2018) is the lack of trained professionals able to perform usability tasks like evaluations, this lack of competence is a major hurdle for the applicability of user centered methods. Therefore, presenting the practitioners with familiar situations and challenges might be a possible way to help practitioners select and identify the right method for the right challenge. Thus, presenting common challenges specifically for the domain the method toolkit is for, might help overcome the hurdle of inaccessibility of traditional UCD methods(Teka et al., 2018) in said domain furthered by lack of knowledge and insight into these types of methods as presented by Ardito et al (2014).

Design principle 4: the toolkit should present examples of solutions to challenges that are readily recognizable by designers

To meet the challenges presented in design principle 3, the resource should have solutions to said challenges too. These examples can be from the empirical insight of the designers themselves or theoretical knowledge if the challenges are known beyond the domain. Specific solutions might be present (Figure 51) but also examples of how other practitioners in the same domain have solved a similar challenge (Figure 51).

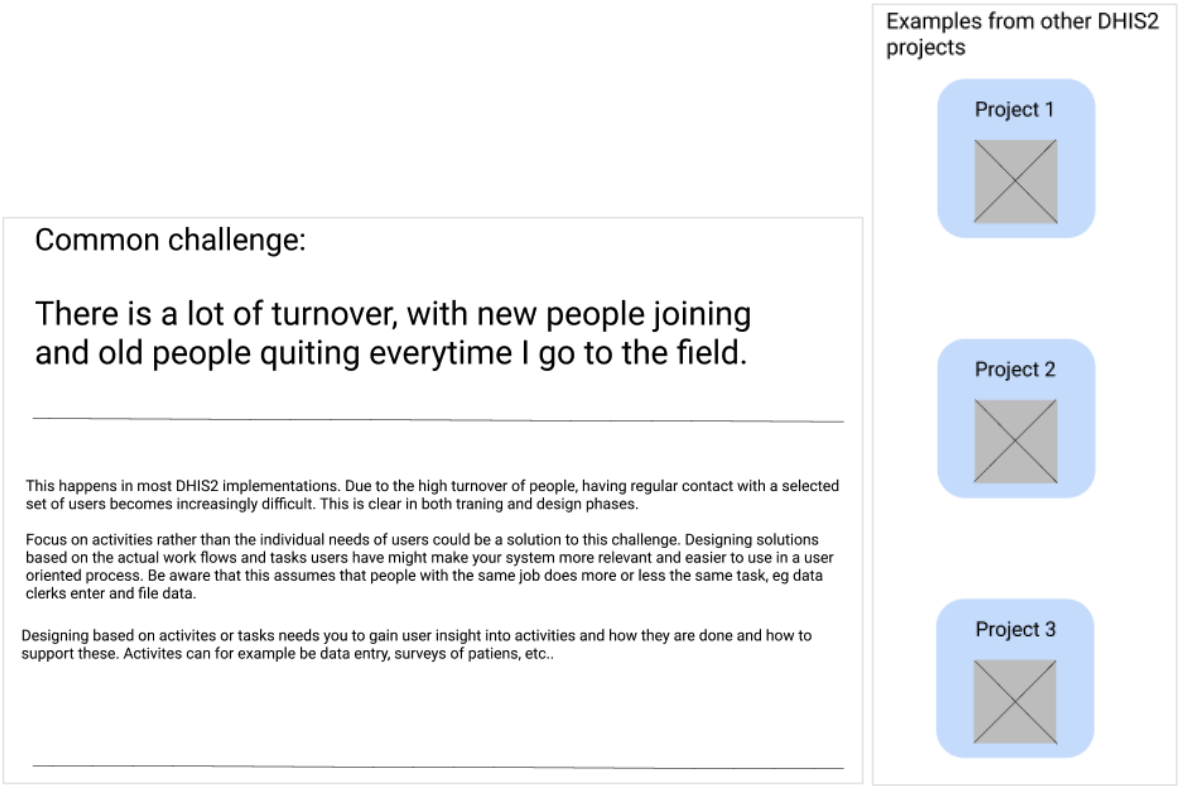


Figure 51: Example of challenge and of examples projects

A significant part of my interviews and observations in the early stages of my project, was focused on not only the challenges that emerged during a software process, but also the solutions. During both interviews and evaluations, there was an interest voiced from the practitioners in different HISP nodes to gain insight into how other HISP nodes have solved similar problems (DP3) and their experiences with different methods. Presenting solutions to the challenges is the "other half" of DP3, thus much of the same arguments can be done for either theoretically.

As with the challenges (DP3), this principle follows the argument of Howarth et al. (2009) concerning scaffolding. Central to his argument too, is that scaffolding functions as competent assistance, helping to achieve a goal thus increasing overall competence. This is echoed by the experiences of Bang et al. "*not just talk about the methods, show the procedures and results"* (Bang et al., 2017, p.184). My study has added to the validity to this statement, and my extension my principle, that presenting domain specific solutions to challenges increases the perceived relevance and accessibility of user centered methods. As one participant during my evaluations,

when asked about the usefulness of the solutions currently present; "*Yes, that would be really nice, because you could use the same approach or elaborate to make it better".* By collection solutions to challenges and "cases" of DHIS2 implementation where these challenges have been met, the toolkit might function at one point as a repository of experiences. Helping to meet Cornet et als (2020) call for more knowledge about practical implementation of UCD methods.

Design principle 5: the toolkit should have a level of abstraction that meets users' mental models

The resource needs to be familiar to the practitioners it is designed for. It should be based on their language, their processes and their methods. However, it is important that if it should include multiple groups/domains/organizations, that a "common" mental model should be established, which in turn needs to have the right level of abstraction. This allows for a more flexible adaption in multiple projects or by multiple organizations and practitioners.

Göransson et al. argues in their investigation of usability tools that *"usability tools, techniques and methods must be integrated and relate to the software-development process"*(Göransson et al., 2003, p. 115). An early feedback during the first evaluation was that the participant did not recognize the process model nor some of the methods presented. However, depending on the users in question, (and as stated in 5.3.1) the processes models that differ from HISP group to HISP group - finding one specific process model might be difficult. As noted during an evaluation; *"[..]...you add validation, verification and everyone into one phase. But they are pretty big ones so that might make it more complicated"*

Familiarity with methods, and by extension processes relevant for the domains practitioners work in, have been hailed as an important aspect of supporting user centered methods(Ardito et al., 2014; Otkjær et al., 2008). Designing a resource that in centered around this familiarly thus is highly important, underlining the need to do significant amount of data gathering insight work.

The principles together adds to Cornet et als (2020) research by being socio-technical (IS) focused, adding how to overcome and meet the complexities of theses contexts. Through these principles ideally there can be more of a methodological "fit", especially for more domain specific challenges for generic software like the dependency on technical flexibility.

The principles

The principles presented in this section are reflective (and prescriptive), as they were mainly developed after design work had started (Möller et al., 2020). The main reason for this being the lack of design principles or guides for development of method toolkits in the literature, thus not being able to build upon principles already present, but rather used to discuss challenges and gaps in the literature. This have made these principles empirically based and extracted from the design itself, one of the types of design principle designing Möller et al. (2020) presents.

The artefact:

The artefact that has been designed in this thesis has been a DHIS2 method toolkit, based on the practices of DHIS2 practitioners. The toolkit itself was evaluated by DHIS2 practitioners themselves and the concept was positivity received. The main goal of this thesis was to establish the utility of such a toolkit to understand if this is a resource that could help support user centered methods when developing and implementing in DHIS2. During one of the evaluations of the utility of the toolkit, one DHIS2 practitioner noted when asked about the potential use of the resource;

> *"If you are new [to DHIS2 implementation], this tool would be good, because you can use the tool for your needs and help all the steps. To for example give a good inception report or organize all the information you get."*

This quote and the findings it represents, aligns well with Bang et als (2017) argument that resources for UCD method are important in organizations where practitioners have limited experience with implementing UCD methods. This is added to as Teka et al (2018) argues that resources might help support local practitioners when using user centered methods by providing them with tangible tools. The DHIS2 method toolkit aims to do just this, and the evaluations points to its utility.

The principles that are derived from the development of the method toolkit allows for a readily accessible choice of methods and techniques as they are based on recognizable challenges and solutions. This might help users of the resource save time and effort finding the right approach for projects and processes, supporting an autonomous method, process and technique choice. Drawing on the lack of perceived relevance of many methods and their lack of fit this might also be

supported through this autonomous choice, as they are not bound to a given process model for example. This is very much in line with the notion of "discount usability", as coined by Nielsen (1995), and can help support the argument that "some usability is better than no usability".

The aim of the toolkit is to promote and support the utilization of user centered methods, thus promoting local autonomy and relevance for UCD, which follows the argument of Piller and Walcher (2006) in their development of toolkits. Through a toolkit, the innovation and use can happen on the users' terms, allowing for little central control of use. This aligns well with the toolkit being a part of an ecosystem and presented my Dittrich (2014). The vendor in this case presents the toolkit, to help foster and support user centered methods, which HISP has as an aim in all their DHIS2 development. This aligns well with how software ecosystem vendors provide resources for the actors that supports the configuration of their software (Dittrich, 2014; Rolland et al., 2018), as methods are a crucial part of the software project.

8.3 Contributions

The contribution of this thesis is threefold, it provides a resource for HISP and DHIS2 practitioners to have their UCD practices supported, it provides a conceptualization for continuous work in the DHIS2 Design Lab both in the shape of principles and the resource itself and it provides an example and principles for the vendor of a software ecosystem in how they can support and promote UCD methods.

The goal of DSR is provide two types contributions, practical and theoretical. The practical contribution of this thesis is a prototype of a method toolkit that can be implemented in the DHIS2 ecosystem. It conceptualizes a resource that practitioners that searches for support in conducting user centered methods can utilize.

The theoretical contribution is to provide design principles concerning how a vendor in a generic software ecosystem can design a resource to support and promote user centered design during implementation and configuration of generic software. As the empirical research and literature in this thesis have presented, the challenges to applying user centered methods are manifold, underlining the importance of understanding contexts and practices. The design principles presented in this thesis helps fill some of the gap identified; yes, user centered methods are

important to reduce usability issues in generic software, but *how* exactly does one do it? The identification of these principles allows for clear guidelines as to how to make a resource to promote and support user centered methods. A resource that has proven utility, and allows for the support of user centered methods.

My study also helps add to empirical insight of challenges and solutions concerning the use of user centered methods when implementing generic software. Providing more explicit insight into the challenges faced by practitioners when using these types of methods, as called for my Cornet et al (2020). These are summarized in Table 9.

Real world problem	Related concerns in academic literature	Addressed by design principle
UCD methods are not perceived as relevant	Some of the promoted UCD methods are note pericived as relevant in the contexts they are promoted in, often due to culture in the organization or cultural context Teka et al(2018), Ardito et al.(2014)	DP2, DP3, DP4, DP5
UCD methods do not fit in the implementation context	Methods are often promoted in a context which is very different from where they originate from. Otkjær et al (2008), Baxter and Summerville (2011) Teka et al (2018)	DP3, DP4, DP2
Perceived high resource cost of UCD	UCD methods are often perceived as highly resource draining, rather than a beneficial activity Ardito et al (2014), Teka et al.(2018), Otkjær et al(2008) Bang et al(2017)	DP2, DP4

There is a lack of" guidance" in the design process	Lack of trained UCD professionals, There is a lack of UCD professionals supporting and leading design processes (Ardito et al(2014), Teka et al (2018)).	DP5, DP4, DP5
UCD is not involved in the call for tenders	UCD methods are not included in calls for tenders, reducing the space for these types of methods (Ardito et al., 2014; Martin et al., 2007)	Needs to be explored further
Communication with users is a hurdle for involving user in the process	--	DP2, DP3, DP4
Often lack of access to end-users in the field makes centering the process on users challenging.	--	DP3, DP4

Table 9: Challenges in literature and empirical findings and design principles

Methodological contribution:

A small methodological contribution was also provided in this thesis. As through the review of DSR articles, very few examples of design principles emerge mainly from empirical data, although Möller et al.(2020) argues that this is one of the main ways to develop design principles. My thesis provides an example of how principles can be derived from empirical data, and used to discuss literature rather than emerge from literature, thus adding to the question Purao et al. asks; *"design principles, where do they all come from"?* (Purao et al., 2020, p. 1).

8.4 Limitations

This thesis has some limitations which are worth to acknowledge. The first being the investigation of practice, but especially through virtual means like online interviews and workshops. As noted throughout this thesis, practices are often bound in tacit knowledge and it is challenging to view oneself and one's practice with the eyes of a stranger. If I had the opportunity to investigate practice more than a few weeks, I could be able to ask questions and immerse myself in a way that was not possible.

Another challenge specifically to the empirical work done online was semantic in nature. It was challenging to know whether or not myself and the participants in my study were talking "about the same thing", when discussing methods, techniques and especially "users". Who the users are and what constitutes a user is highly different from person to person, project to project and group to group. If I had the opportunity to immerse myself in the context and practices taking place in the HISP groups, I might have had the possibility to ask questions about who I perceive are the users and what I perceive are methods to make sure we were discussing the same thing. This was made more difficult by infrastructural issues with virtual empirical work, like bad internet connections and equipment making the sound and picture hard to understand at times.

Further it is worth to note that the evaluations of the artefact only were done within one HISP group, only reflecting their views. The HISP Mozambique group however has responsibility for many other Portuguese countries in Africa and are frequent participations in international conferences and exchanges, giving the members some insight into the practices of other HISP groups as well.

Finally, the artefact has not been tested in a "real" DHIS2 implementation project, making it challenging to assess how it would be used during projects or in project planning. It might be different as the evaluations only centered on demonstrations and scenarios from previous or fictive projects. This was the intent for the original, pre-Covid 19 project, as I was going to be present for the introduction of an artefact into the context of use (HISP Mozambique). This could have given more opportunity to test specific aspects of the toolkit, like specific methods for specific projects. The practitioners could also then be included to larger extent, thus have more power in the design and development of a resource where they are the target users.

Chapter 9: Conclusion

This thesis has explored how user centered methods can be supported and promoted when implementing generic software in a generic software ecosystem. This was explored through the development of a method toolkit specifically for user centered design in the DHIS2 ecosystem. Through a yearlong design science research project, I collaborated with practitioners by both virtual and physical means, gaining a richer knowledge and insight into their practices as well as challenges to utilizing user centered design methods when configuring and implementing the generic software DHIS2.

The creation of this artefact (the DHIS2 method toolkit) was constructed through applying design science research as a methodology, therefore the main theoretical contribution of this thesis are five design principles for how a vendor of a software ecosystem can create a toolkit can support user oriented design and what this toolkit should be. These are:

1. DP1: The toolkit should be mutable
2. DP2: The toolkit should present a diversity of methods, techniques and processes
3. DP3: the toolkit should present readily recognizable examples of common challenges
4. DP4: the toolkit should present examples of solutions to challenges that are readily recognizable by designers
5. DP5: the toolkit should have a level of abstraction that readily meets users' mental models

These principles can be leveraged upon by other vendors in a software ecosystem, giving some guidance as to how to construct a resource for the support of user centered design methods in a software ecosystem, and what such a resource could be. It also provides more insight into how UCD methods are currently being conducted in a generic software ecosystem like the one based on DHIS2.

The prototypes of the DHIS2 method toolkit was evaluated with DHIS2 practitioners, and might be one contribution and solution to the practical challenges reported from the DHIS2 practitioners themselves with conducting user oriented methods. Like lack of fit with often promoted UCD methods in their current practices.

9.1 Future Work

As noted throughout this thesis, this thesis was the beginning of a larger project in the DHIS2 Design Lab, and the artefact created in this thesis was mainly a conceptual design endeavor. This leads to future work specifically centered on the toolkit and its evolution. There should be more work investigating the use of the toolkit in "real" projects, where the toolkit is used in a specific project setting, as it is intended. This is currently being planned and taking place in the DHIS2 Design Lab. As the aim of the toolkit is to support multiple facets of the implementation and configuration process, this would prove highly useful. This also allows the toolkit to be tested and utility to be established across time in longer projects and across multiple types of projects.

The form of content and the form of the toolkit itself should also be iterated on, investigating accessibility, utility and usability of the toolkit itself. With questions such as, what form should methods, processes and techniques have when presented to practitioners? For the toolkit, further to be used as a "repository of experiences" from the DHIS2 ecosystem, more collections of projects, their processes and challenges should be collected and gathered in the toolkit to guide and help DHIS2 practitioners across the ecosystem.

Another aspect that could be interesting to explore, that I was unable to do, is the process of establishing mandate for projects. As noted in this thesis and in the literature, the lack of UCD in calls for tenders for projects have an effect on to what extend these types of methods are used. Therefore, exploring with practitioners what is needed to include more UCD in these calls for tenders and how to actively support this process needs more work.

The presented design principles should be tested with a wider array of practitioners, and further be iterated and refined. Their accessibility and validity should be tested in multiple instances and maybe more principles emerge or some might need to be removed as more insight and knowledge is gained.

Concluding remarks

This thesis has tried to explore the complex world of socio-technical methods and the practices they exist in. Methods have a uniquely human aspect to them and will always serve as a tool to meet an end, therefore, as I have tried to explore and promote through my design principles, the

importance of including the people that have a say in these methods and practices are crucial. A resource like this is only of utility and use if they target users see the relevance of the artefact itself. Taking the time to listen to practitioners, listen to their needs, their challenges and their solutions is therefore crucial when creating something you want to encourage them all to use.

REFERENCES

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

Bakke, S., & Bratteteig, T. (2015). The Closer the Better: Effects of Developer-User Proximity for Mutual Learning. In M. Kurosu (Ed.), *Human-Computer Interaction: Design and Evaluation* (pp. 14–26). Springer International Publishing.

Bang, K., Kanstrup, M. A., Kjems, A., & Stage, J. (2017). Adoption of UX Evaluation in Practice: An Action Research Study in a Software Organization. In R. Bernhaupt, G. Dalvi, A. Joshi, D. K. Balkrishan, J. O'Neill, & M. Winckler (Eds.), *Human-Computer Interaction – INTERACT 2017* (pp. 169–188). Springer International Publishing. https://doi.org/10.1007/978-3-319-68059-0_11

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

Baskerville, R., Baiyere, A., University of Turku, Finland, Gergor, S., Australian National University, Hevner, A., University of South Florida, Rossi, M., & Aalto University. (2018). Design Science Research Contributions: Finding a Balance between Artifact and Theory. *Journal of the Association for Information Systems*, *19*(5), 358–376. https://doi.org/10.17705/1jais.00495

Baskerville, R., & Pries-Heje, J. (2010). Explanatory Design Theory. *Business & Information Systems Engineering*, *2*(5), 271–282. https://doi.org/10.1007/s12599-010-0118-4

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

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

Braa, J., & Sahay, S. (2012). Health information systems programme: Participatory design within the HISP network. In *Routledge International Handbook of Participatory Design* (pp. 235–256). Taylor and Francis. http://web.a.ebscohost.com.ezproxy.uio.no/ehost/ebookviewer/ebook/bmxlYmtfXzQ5NTA1N19fQU41?sid=7bc1bb76-78d7-4d0b-983b-9317e4d0c61d@sessionmgr4007&vid=0&format=EB&rid=1

Braun, V., & Clarke, V. (2012). Thematic analysis. In H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, & K. J. Sher (Eds.), *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological.* (pp. 57–71). American Psychological Association. https://doi.org/10.1037/13620-004

Brhel, M. (2015). Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, 19.

Brown, T. (2008). Design Thinking. *Harvard Business Review*, 11.

Carlsson, S. A., Henningsson, S., Hrastinski, S., & Keller, C. (2011). Socio-technical IS design science research: Developing design theory for IS integration management. *Information Systems and E-Business*

Management, *9*(1), 109–131. https://doi.org/10.1007/s10257-010-0140-6

Chafi, B. (2020). *Contextual user research methods for eliciting user experience insights in workplace studies*. 12.

Choma, J., Zaina, L. A. M., & Silva, T. S. D. (2015). Towards an Approach Matching CMD and DSR to Improve the Academia-Industry Software Development Partnership: A Case of Agile and UX Integration. *2015 29th Brazilian Symposium on Software Engineering*, 51–60. https://doi.org/10.1109/SBES.2015.18

Cockton, G., & Woolrych, A. (2002). Sale must end: Should discount methods be cleared off HCI's shelves? *Interactions*, *9*(5), 13–18. https://doi.org/10.1145/566981.566990

Cornet, V. P., Toscos, T., Bolchini, D., Ghahari, R. R., Ahmed, R., Daley, C., Mirro, M. J., & Holden, R. J. (2020). Untold Stories in User-Centered Design of Mobile Health: Practical Challenges and Strategies Learned From the Design and Evaluation of an App for Older Adults With Heart Failure. *JMIR MHealth and UHealth*, *8*(7), e17703. https://doi.org/10.2196/17703

Cvijikj, I. P., & Michahelles, F. (2011). The Toolkit Approach for End-user Participation in the Internet of Things. In D. Uckelmann, M. Harrison, & F. Michahelles (Eds.), *Architecting the Internet of Things* (pp. 65–96). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19157-2_4

DHIS2. (2021, March 30). *Bangladesh uses DHIS2 to manage immunization of 35+ million children in their MR mass campaign*. DHIS2. https://dhis2.org/bangladesh-immunization-campaign/

DHIS2. (n.d.-a). *About DHIS2*. DHIS2. Retrieved May 8, 2021, from https://dhis2.org/about/

DHIS2. (n.d.-b). *Academy*. DHIS2. Retrieved May 16, 2021, from https://dhis2.org/academy/

DHIS2. (n.d.-c). *HISP Network*. DHIS2. Retrieved May 12, 2021, from https://dhis2.org/hisp-network/

DHIS2. (n.d.-d). *Home—DHIS2 Documentation*. Retrieved May 16, 2021, from https://docs.dhis2.org/en/home.html

DHIS2. (2020, November 27). *DHIS2 tracker e-Registry in Palestine*. DHIS2. https://dhis2.org/palestine-tracker-user-story/

DHIS2. (2020, November 30). *The Nature Conservancy uses DHIS2 to support conservation in Eastern Africa*. DHIS2. https://dhis2.org/eastern-africa-tnc-conservation/

DHIS2. (2021, March 30). *Bangladesh uses DHIS2 to manage immunization of 35+ million children in their MR mass campaign*. DHIS2. https://dhis2.org/bangladesh-immunization-campaign/

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

Dittrich, Yvonne. (2014a). 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, Yvonne, Rönkkö, K., Eriksson, J., Hansson, C., & Lindeberg, O. (2008b). Cooperative method

development: Combining qualitative empirical research with method, technique and process improvement. *Empirical Software Engineering*, *13*(3), 231–260. https://doi.org/10.1007/s10664-007-9057-1

Edwards, R., & Holland, J. (2013). *What is Qualitative Interviewing?* (Library of Congress Cataloging-in-Publication Data). Bloomsbury Academic.

Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: The boundary resources model: Control and contribution in third-party development. *Information Systems Journal*, *23*(2), 173–192. https://doi.org/10.1111/j.1365-2575.2012.00406.x

Goldkuhl, G., & Karlsson, F. (2020). Method Engineering as Design Science. *Journal of the Association for Information Systems*, *21*(5), 1237–1278. https://doi.org/10.17705/1jais.00636

Göransson, B., Gulliksen, J., & Boivie, I. (2003). The usability design process – integrating user-centered systems design in the software development process. *Software Process: Improvement and Practice*, *8*(2), 111–131. https://doi.org/10.1002/spip.174

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337–355. https://doi.org/10.25300/MISQ/2013/37.2.01

Grudin, J. (2009). AI and HCI: Two Fields Divided by a Common Focus. *AI Magazine*, *30*(4), 48. https://doi.org/10.1609/aimag.v30i4.2271

Gulliksen, J., Goransson, B., Boivie, I., Persson, J., & Blomkvist, S. (2003). 2 KEY PRINCIPLES FOR USER-CENTRED SYSTEMS DESIGN. *2005*, 20.

Hartswood, M., Procter, R., Slack, R., Soutter, J., Voß, A., & Rouncefield, M. (2002). *The Benefits of a Long Engagement: From Contextual Design to The Co-realisation of Work Affording Artefacts*. 4.

Heer, J., Card, S. K., & Landay, J. A. (2005). prefuse: A toolkit for interactive information visualization. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '05*, 421. https://doi.org/10.1145/1054972.1055031

Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In A. Hevner & S. Chatterjee (Eds.), *Design Research in Information Systems: Theory and Practice* (pp. 9–22). Springer US. https://doi.org/10.1007/978-1-4419-5653-8_2

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. JSTOR. https://doi.org/10.2307/25148625

HISP. (n.d.). *Health Information Systems Programme (HISP)—Department of Informatics*. Retrieved April 23, 2021, from https://www.mn.uio.no/ifi/english/research/networks/hisp/index.html

Holtzblatt, K., Beringer, J., & Baker, L. (2005). *Rapid User Centered Design Techniques: Challenges and Solutions*. 2.

Howarth, J., Smith-Jackson, T., & Hartson, R. (2009). Supporting novice usability practitioners with usability engineering tools. *International Journal of Human-Computer Studies*, *67*(6), 533–549. https://doi.org/10.1016/j.ijhcs.2009.02.003

IDEO. (n.d.). *Design Method Toolkit by the Digital Society School*. Retrieved April 13, 2021, from https://toolkits.dss.cloud/design/

Isaacs, E. (2012). The Value of Rapid Ethnography. In *Advancing Ethnography in Corporate Environments: Challenges and Emerging Opportunities* (1st ed.). Routledge. https://doi.org/10.4324/9781315435459

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

Kane, D. (2003). Finding a place for discount usability engineering in agile development: Throwing down the gauntlet. *Proceedings of the Agile Development Conference, 2003. ADC 2003*, 40–46. https://doi.org/10.1109/ADC.2003.1231451

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

Li, M. (2019a). *Making Usable Generic Software—The Platform Appliances Approach*.

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

Li, M., & Nielsen, P. (2019). 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., Procter, R., Mariani, J., & Rouncefield, M. (2007). Working the contract. *Proceedings of the 2007 Conference of the Computer-Human Interaction Special Interest Group (CHISIG) of Australia on Computer-Human Interaction: Design: Activities, Artifacts and Environments - OZCHI '07*, 241. https://doi.org/10.1145/1324892.1324945

Mithun, Ahamed. M., Mithun, Ahamed. M., & Yafooz, Wael. M. S. (2018). Extended User Centered Design (UCD) Process in the Aspect of Human Computer Interaction. *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, 1–6. https://doi.org/10.1109/ICSCEE.2018.8538388

Möller, F., Guggenberger, T. M., & Otto, B. (2020). Towards a Method for Design Principle Development in Information Systems. In S. Hofmann, O. Müller, & M. Rossi (Eds.), *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry* (pp. 208–220). Springer International Publishing. https://doi.org/10.1007/978-3-030-64823-7_20

Mumford, E. (2006). The story of socio-technical design: Reflections on its successes, failures and potential. *Information Systems Journal*, *16*(4), 317–342. https://doi.org/10.1111/j.1365-2575.2006.00221.x

Nielsen, J. (1995). Applying discount usability engineering. *IEEE Software*, *12*(1), 98–100. https://doi.org/10.1109/52.363161

Nielsen, Jakob. (n.d.). *Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier*. 18.

Norman, D. A., & Draper, S. W. (1986). *User centered system design: New perspectives on human-computer interaction*. Lawrence Erlbaum.

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

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

Piller, F. T., & Walcher, D. (2006). Toolkits for idea competitions: A novel method to integrate users in new product development. *R&D Management*, *36*(3), 307–318. https://doi.org/10.1111/j.1467-9310.2006.00432.x

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

Preece, J., Sharp, H., & Yvonne, R. (2015). *Interaction design: Beyond human-computer interaction* (4th ed.). Wiley.

Purao, S., Kruse, L. C., & Maedche, A. (2020). The Origins of Design Principles: Where do… they all come from? In S. Hofmann, O. Müller, & M. Rossi (Eds.), *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry* (Vol. 12388, pp. 183–194). Springer International Publishing. https://doi.org/10.1007/978-3-030-64823-7_17

Rolland, K. H., Mathiassen, L., & Rai, A. (2018). Managing Digital Platforms in User Organizations: The Interactions Between Digital Options and Digital Debt. *Information Systems Research*, *29*(2), 419–443. https://doi.org/10.1287/isre.2018.0788

SAP. (n.d.). Process Tools. *SAP AppHaus*. Retrieved April 13, 2021, from https://experience.sap.com/designservices/tools/process

Seidel, S., Chandra Kruse, L., Székely, N., Gau, M., & Stieger, D. (2018). Design principles for sensemaking support systems in environmental sustainability transformations. *European Journal of Information Systems*, *27*(2), 221–247. https://doi.org/10.1057/s41303-017-0039-0

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

Service Design Tools. (n.d.). *Tools | Service Design Tools*. Retrieved May 10, 2021, from https://servicedesigntools.org/tools.html

Simonsen, J., & Robertson, T. (2012). *Routledge International Handbook of Participatory Design*. https://doi.org/10.4324/9780203108543

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

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

Teka, D., Dittrich, Y., & Kifle, M. (2018). Adapting lightweight user-centered design with the scrum-based development process. *Proceedings of the 2018 International Conference on Software Engineering in Africa - SEiA '18*, 35–42. https://doi.org/10.1145/3195528.3195530

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

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. https://doi.org/10.1057/ejis.2014.36

Vilpola, I. H. (2008). A method for improving ERP implementation success by the principles and process of user-centred design. *Enterprise Information Systems*, *2*(1), 47–76. https://doi.org/10.1080/17517570701793848

Visser, F. S., Stappers, P. J., van der Lugt, R., & Sanders, E. B.-N. (2005). Contextmapping: Experiences from practice. *CoDesign*, *1*(2), 119–149. https://doi.org/10.1080/15710880500135987

Wareham, J., Fox, P. B., & Giner, J. L. C. (2014). Technology Ecosystem Governance. *Organization Science*, 22.

Wong, C. Y., Chu, K., & Pauzi, M. A. M. (2016). Advocating UX practice in industry: Lessons learnt from UX innovate bootcamp. *2016 4th International Conference on User Science and Engineering (i-USEr)*, 204–209. https://doi.org/10.1109/IUSER.2016.7857961